NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

THESIS

IMPLEMENTING AND BOUNDING A CASCADE HEURISTIC FOR LARGE-SCALE OPTIMIZATION

by

Katherine H. Guthrie

June 2017

Thesis Advisor: Robert F. Dell
Second Reader: Gerald G. Brown

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE			*Form Approved OMB No. 0704-0188*
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington DC 20503.			
1. AGENCY USE ONLY ***(Leave blank)***	**2. REPORT DATE** June 2017	**3. REPORT TYPE AND DATES COVERED** Master's thesis	
4. TITLE AND SUBTITLE IMPLEMENTING AND BOUNDING A CASCADE HEURISTIC FOR LARGE-SCALE OPTIMIZATION			**5. FUNDING NUMBERS**
6. AUTHOR(S) Katherine H. Guthrie			
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey, CA 93943-5000			**8. PERFORMING ORGANIZATION REPORT NUMBER**
9. SPONSORING /MONITORING AGENCY NAME(S) AND ADDRESS(ES) N/A			**10. SPONSORING / MONITORING AGENCY REPORT NUMBER**
11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government. IRB number ____N/A____.			
12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release. Distribution is unlimited.			**12b. DISTRIBUTION CODE**

13. ABSTRACT (maximum 200 words)

A cascade heuristic appeals when we are faced with a monolithic optimization model exhibiting more decision variables and/or constraints than can be accommodated by computers and/or optimization software available. This thesis studies the implementation and bounding of a cascade heuristic by using the integer linear program implementations of two applications, a production model (PM) and the USMC Hornet Assignment Sundown Model (HASMa). While the solutions for PM are within 5% of the optimal solution for a wide variety of cascade heuristic implementations, the solutions for HASMa deviate, in some cases, by over 99% of the optimal solution. To provide a metric for the quality of a cascade heuristic solution, we produce a lower bound for the optimal objective function value by aggregating segments of each model's periods. For PM, the aggregated models produce lower bounds all within 2% of the optimal objective function value. For HASMa, the lower bounds can be up to 50% from the optimal objective function value but are within 10% of optimal when the aggregation includes just one-third of the periods. In both cases, finding a lower bound for the optimal objective function value provides significant insight to the quality of the cascade heuristic solution.

14. SUBJECT TERMS cascade heuristic, cascade optimization, F/A-18, production model, bounding objective function values			**15. NUMBER OF PAGES** 79
			16. PRICE CODE
17. SECURITY CLASSIFICATION OF REPORT Unclassified	**18. SECURITY CLASSIFICATION OF THIS PAGE** Unclassified	**19. SECURITY CLASSIFICATION OF ABSTRACT** Unclassified	**20. LIMITATION OF ABSTRACT** UU

NSN 7540-01-280-5500 Standard Form 298 (Rev. 2-89) Prescribed by ANSI Std. 239-18

THIS PAGE INTENTIONALLY LEFT BLANK

Approved for public release. Distribution is unlimited.

IMPLEMENTING AND BOUNDING A CASCADE HEURISTIC FOR LARGE-SCALE OPTIMIZATION

Katherine H. Guthrie
Captain, United States Marine Corps
B.S., United States Naval Academy, 2011

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN OPERATIONS RESEARCH

from the

NAVAL POSTGRADUATE SCHOOL
June 2017

Approved by: Robert F. Dell
Thesis Advisor

Gerald G. Brown
Second Reader

Patricia A. Jacobs
Chair, Department of Operations Research

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

A cascade heuristic appeals when we are faced with a monolithic optimization model exhibiting more decision variables and/or constraints than can be accommodated by computers and/or optimization software available. This thesis studies the implementation and bounding of a cascade heuristic by using the integer linear program implementations of two applications, a production model (PM) and the USMC Hornet Assignment Sundown Model (HASMa). While the solutions for PM are within 5% of the optimal solution for a wide variety of cascade heuristic implementations, the solutions for HASMa deviate, in some cases, by over 99% of the optimal solution. To provide a metric for the quality of a cascade heuristic solution, we produce a lower bound for the optimal objective function value by aggregating segments of each model's periods. For PM, the aggregated models produce lower bounds all within 2% of the optimal objective function value. For HASMa, the lower bounds can be up to 50% from the optimal objective function value but are within 10% of optimal when the aggregation includes just one-third of the periods. In both cases, finding a lower bound for the optimal objective function value provides significant insight to the quality of the cascade heuristic solution.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

LIST OF FIGURES

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF TABLES

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF ACRONYMS AND ABBREVIATIONS

GAMS	Generalized Algebraic Modeling System
HASM	Hornet Assignment Sundown Model
HASMa	Hornet Assignment Sundown Model (Revised)
HFH	High Flight Hour Extension
ILP	Integer Linear Program
KPS	Kellogg Planning System
NAVAIR	Naval Air Systems Command
NSL	Non-aggregated Segment Length
PM	Production Model
PMI	Planned Maintenance Interval
RMIP	Relaxed Mixed Integer Program
RSRP	Rolling Stock Rescheduling Program
SFGM	Security Force Generation Model
SLE	Service Life Extension
TCM	Training Capability Model
USMC	United States Marine Corps

THIS PAGE INTENTIONALLY LEFT BLANK

EXECUTIVE SUMMARY

A cascade heuristic appeals when we are faced with a monolith optimization model exhibiting more decision variables and/or constraints than can be accommodated by computers and/or optimization software available. A cascade heuristic attempts to indirectly solve a monolithic optimization problem by solving a sequence of sub-problems, or blocks, and conveying information from block to block. Each block may include, say, a subset of all time periods, products, regions, or other related problem components, while all constraints not in the block are relaxed, and all decision variables not in the block are fixed. As each successive cascade block is solved, a subset of the resulting variable values are fixed before another block is solved. Most typically, the cascade involves sweeping through blocks of monolith constraints and variables once to indirectly assemble a solution to the monolith. Because cascade blocks are formed by, at once, fixing variables (a restriction) and relaxing constraints (a relaxation), there is no direct relationship between the overall monolith solution achieved by a cascade. In fact, the cascade heuristic may myopically assemble an infeasible solution for a feasible monolith, or an unbounded solution for a bounded monolith. Despite these hazards, a cascade heuristic frequently results in useful solutions to otherwise intractable monoliths.

This thesis considers the most typical application of a cascade heuristic when there are discrete time periods that extend over either a finite or an infinite time horizon. For both horizon types, a cascade heuristic breaks the monolith into overlapping subsets of the time periods defined by windows. The heuristic then solves multiple iterations of the problem over a succession of windows. The first iteration (the first block) considers only the periods included in the first window with all other constraints relaxed and all other variables fixed. Based upon the solution obtained, variable values within a subset of the window are fixed. The window then slides forward to the end of an advance (the parameter that specifies how far forward the window progresses each iteration) to establish the second window. The cascade heuristic solves the second window, fixes variable values based on the obtained solution, and continues in a similar fashion.

This thesis looks at two separate applications and their integer linear program implementations. We use these applications to analyze the impact on solution quality when varying the implementation of the cascade heuristic and when developing lower bounds for the optimal objective function value. We examine a production model (PM) and a revised Hornet Assignment Sundown Model (HASMa). This thesis also considers variations of each model to produce lower bounds of the optimal objective function value.

The cascade heuristic produces high-quality solutions for PM for all lengths of window and advance. For PM, there is no conclusive discovery about how long an advance should be for a given window length. In the PM cascade heuristic results, we find that longer advances do not nessessarily equate to worse quality solutions. Also for PM, there is no monotonic trend in solution quality when the window length advances.

The quality of solutions produced through the cascade of HASMa are quite different from the quality of solutions produced in the PM cascade. For HASMa, certain lengths of window and advance produce solutions with a nearly 100% deviation from the optimal solution. This results from the inability of HASMa to easily recover from poor decisions made early in the model. Cascade heuristic solutions from PM echo these results when we modify PM's data to include a significant demand spike in a late period (period ten) that no window will discover until near the end of the cascade. When this demand spike is included, a window length of seven periods is required to produce a cascade heuristic solution within 2% of the optimal solution.

To provide a metric to judge the quality of a cascade heuristic solution without solving the monolith, we develop a method for producing lower bounds to the optimal objective function value. To do this, we solve a new integer linear program with aggregated constraints for time periods both early and late in the model. The objective function value that results is a lower bound of the monolith's objective function value. The quality of the lower bound depends on the number of periods included in the aggregations. For PM, the lower bounds are all within 2% of the optimal objective function value, even when including almost all periods in the aggregations. For HASMa, the lower bounds produced can be up to 50% from the optimal objective function value

but are within 10% of optimal when just one-third of the periods are included in the aggregation. In both cases, aggregating the models to find a lower bound for the optimal objective function value provides significant insight into the quality of the cascade heuristic solution.

The cascade results found from each of the two models are similar in some aspects and vastly different in others, which highlights the complexity of the cascade technique and the difficulty in making any broad conclusions regarding the application of a cascade heuristic. However, the results found in this thesis provide insight into the implementation and bounding of a cascade heuristic that could be applicable for other models. The results of the cascade of each model support the recommendation that, for any model using a cascade heuristic, one should use the longest window length that is computationally feasible to produce higher-quality solutions. Less conclusive evidence exists to back a strong statement regarding the length of the advance. Results from HASMa suggest that shorter advances typically yield higher-quality solutions, but this is not always the case for PM. Additionally, we can make no specific statements regarding how long the window length should be with respect to the time horizon of the model to guarantee any quality of solution. The results from this thesis suggest the importance of determining a lower bound for any model solved using a cascade heuristic. For both models examined, the aggregation technique produces a bound for the model that, when used with the results from the cascade heuristic, give the user a clear measure on the potential quality of the solution.

THIS PAGE INTENTIONALLY LEFT BLANK

ACKNOWLEDGMENTS

I would like to thank my family, particularly my husband, Pierce, and my parents for their support over the last two years. It has been a family effort to get both Pierce and me to graduation, and I am so thankful that we are finally here. Pierce, thank you for your patience, understanding, and commitment to our family. Mom and Dad, thank you for all of your years of love and support. Many thanks to my daughter, Lelia, for your silliness and sweetness, and for reminding me of what is most important in life.

I would also like to thank my advisors, Dr. Dell and Dr. Brown, for your support while writing this thesis. I have learned a tremendous amount over the past year through your guidance, and I am truly grateful for all of the time and energy spent helping me through the process.

THIS PAGE INTENTIONALLY LEFT BLANK

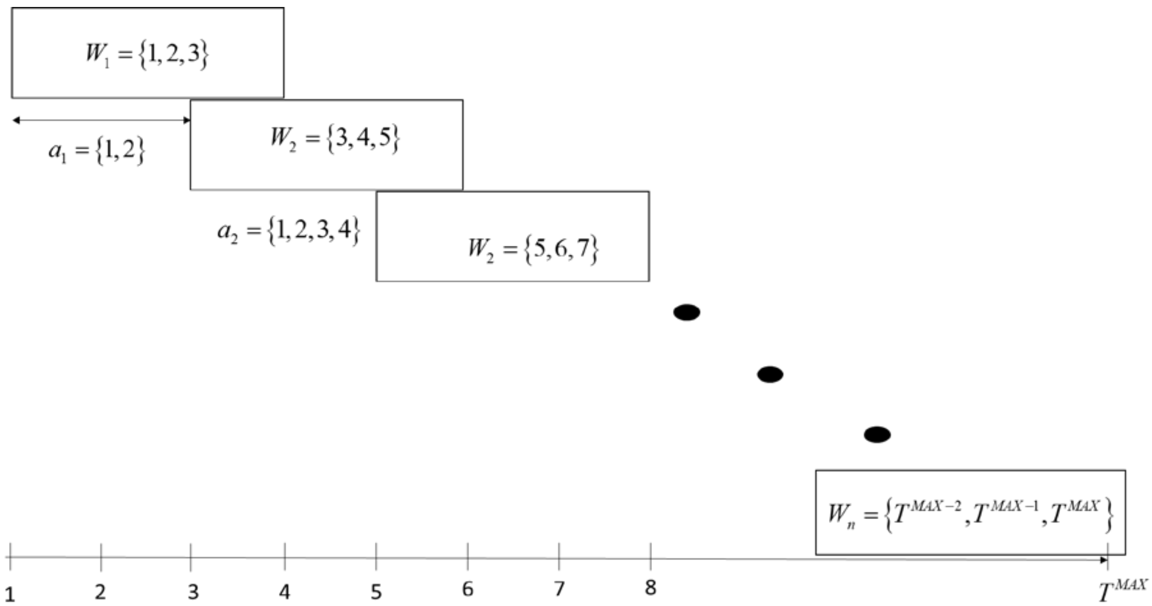
I. INTRODUCTION

A. OVERVIEW

The Oxford English Dictionary defines a cascade as, “a process whereby something, typically information or knowledge, is successively passed on” (“Cascade,” n.d.). A cascade heuristic, true to this definition, is an optimization-based technique that attempts to indirectly solve a monolithic optimization problem by solving a sequence of sub-problems, or blocks, and conveying information from block to block. Each block may include, say, a subset of all time periods, products, regions, or other related problem components, while all constraints not in the block are relaxed, and all decision variables not in the block are fixed. As each successive cascade block is solved, a subset of the resulting variable values are fixed before another block is solved. Most typically, the cascade heuristic involves sweeping through blocks of monolith constraints and variables once, with block solutions used to indirectly assemble a solution to the monolith. Because cascade blocks are formed by, at once, fixing variables (a restriction) and relaxing constraints (a relaxation), there is no direct relationship between the overall monolith solution achieved by a cascade. In fact, the cascade heuristic may myopically assemble an infeasible solution for a feasible monolith, or an unbounded solution for a bounded monolith. Despite these hazards, a cascade heuristic frequently results in useful solutions to otherwise intractable monoliths.

This thesis considers the most typical application of a cascade heuristic when there are discrete time periods that extend over either a finite or an infinite time horizon. In the case of finite horizons, the time periods of the problem make up the set $t = 1, 2, ...T^{MAX}$. For both horizon types, a cascade heuristic breaks the monolith into overlapping subsets of the time periods defined by active windows. The heuristic then solves multiple active iterations of the problem. The first iteration (the first block) considers only the periods included in the first window with all other constraints relaxed and all other variables fixed. Based upon the solution obtained, variable values within a subset of the window are fixed. The active window then slides forward to the end of an “advance,” the parameter that specifies how far forward the window progresses each

iteration, to establish the second window. This cascade heuristic solves the second window, fixes variable values based on the obtained solution, and continues in a similar fashion, as shown in Figure 1. Table 1 provides a summary of the nomenclature we use for the cascade heuristic in this thesis. Cascade optimization also goes by the name of rolling (or sliding) horizon optimization, and this thesis uses one of these other terms when it provides consistency with a published reference.



Each successive window consists of enforced constraints and relaxed variables, while all other constraints are relaxed and all other variables are fixed.

Figure 1. Overlapping Windows of a Cascade Heuristic.
Adapted from Baker (1997).

Table 1. Nomenclature Used to Describe a Cascade Heuristic

Description	Nomenclature
Set of all time periods within the problem	$t = 1, 2, \ldots, T^{MAX}$
Set of all iterations required to solve problem	$i = 1, 2, \ldots, n$
Window: set of time periods solved in iteration i	W_i
Advance: set of time periods fixed for iteration i based on solutions from the previous iterations	a_i

Brown, Graves, and Ronen (1987) first coined the term "cascade optimization" when they used the technique to solve an ocean transport model to schedule shipping of crude oil. However, the theory and practical application of a cascade heuristic has existed for much longer. Dantzig (1959) discusses one early implementation of the technique to solve a motor-steel-tool model, a large (by 1950s standards) multistage linear program. To solve, Dantzig found an optimal solution for $t = 1$ and then used that solution as the initial conditions for $t = 2$. The process continued period-by-period until the monolith was heuristically solved in its entirety.

Although cascade heuristics have been used to solve optimization problems for a long time, there is currently little guidance on how to determine the length of either the window or the advance, for those cascades consisting of only a single, sequential pass of windows. As optimization uses all of the resources available in a given time period, choosing a window that is too short can result in a myopic model fraught with end effects, the term used to describe premature utilization of resources in an early time period, or unrealistic prescriptions in later time periods caused by no visible future. Alternatively, longer windows can result in longer solve times.

B. REASONS FOR THIS STUDY

A cascade heuristic has two problems: 1) the heuristic typically offers no bounds on solution quality; and 2) there is little guidance on how to select the lengths for window and advance. This thesis provides insight on these two problems by analyzing the effect

of varying the length of the window and advance on solution quality, as well as developing and analyzing methods for producing lower bounds to the optimal objective function value of the seminal problem monolith, which this heuristic may never solve completely.

C. THESIS SCOPE AND ORGANIZATION

This thesis comprises five chapters. Chapter I introduces cascade optimization and the shortcomings of the heuristic. Chapter II includes a literature review of related research. Chapter III describes formulations of the two models studied in this thesis, a production model (PM) and a revised Hornet Assignment Sundown Model (HASMa), as well as formulations for an aggregated version of each model. Chapter IV describes the computer implementation of the models and the analysis of results. Chapter V presents conclusions and discusses recommendations for related research.

II. LITERATURE REVIEW

There is extensive literature relating to rolling decision horizons, and the development and implementation of a cascade heuristic. The majority of this literature falls in one of three broad categories: the presentation of applications that use a cascade heuristic and the selection of the window length, the correction of end effects associated with implementing a cascade heuristic, and the convergence of what will be introduced and defined as primal and dual equilibrium approximations.

A. PREVIOUS CASCADE HEURISTIC WORK

There exists an impressive collection of diverse applications solved, at least approximately, by using a cascade heuristic. This section presents a sample of these applications chosen for their diversity and relevance to this thesis.

Brown, Keegan, Vigus, and Wood (2001) implement a rolling-horizon heuristic when solving the Kellogg Planning System (KPS), an infinite-horizon production-planning model that, when published, had been utilized by The Kellogg Company for more than a decade to optimize production, inventory, and distributional decisions. KPS solved a 30-week planning horizon at the beginning of every quarter to develop a long-term production plan. However, the large variability in the product demand data used in KPS made accepting production decisions 30 weeks in advance very risky. As a result, Kellogg planners ran KPS at the beginning of every week to reaffirm production decisions for the close future to mitigate their risk. KPS uses sliding time windows to not only make the time horizon of the problem feasible to solve, but also to take advantage of the narrow time window to induce desired results without having to add additional constraints, and therefore increase the size of the model. For example, the length of the windows used in KPS create a myopic model that cannot see beyond the six-month time horizon to ensure that KPS does not hold products to satiate a demand that occurs beyond the end of their shelf life.

Nielsen, Kroon, and Maróti (2012) solve the Rolling Stock Rescheduling Problem (RSRP), a model that schedules changes to the Norwegian Railways train schedule after a

disruption occurs. RSRP uses a rolling horizon heuristic to reduce computation time to allow real-time solving of the model and to introduce uncertainty into the model. As disruptions are unplanned events, such as damage occurring to a section of railway between stations, there is no way of knowing when normal operations can resume. The length of the time horizon is initially set to the best guess of when normal operations will resume, and planners solve RSRP periodically as railway administrators obtain new information. RSRP then extends or shortens the length of the time horizon as new information dictates. Nielsen et al. (2012) held all of variables constant and varied the length of the window of RSRP from two to five hours in fifteen-minute increments. They found that, in general, longer windows produced higher quality solutions. Brown et al. (1987) had similar findings regarding window length with their ocean transport model by varying window length from 10 to 80 days and comparing the model results. They found that while their model solved to integer optimality for each length window, shortening the window significantly improved model solve times.

Miller et al. (2017) develop the Training Capability Model (TCM) to allocate students from each class of the United States Naval Nuclear Propulsion Training Program to one of the four Nuclear Power Training Units, where they complete their training and certification. Additionally, TCM prescribes weekly staff instructor assignments, student watch-standing assignments, and off-watch training. While the typical TCM planning horizon of two-to-four years can be solved as a monolith, by implementing a cascade heuristic with a window length of 1.5 years and advance of 0.5 years, the solve time can be significantly reduced. The authors report limited computational experience with cascade solutions within 1% of the monolith’s objective function value. While the interface of TCM allows the planner to decide whether to run the model using the rolling horizon or just solve the monolith outright, users have found they generally prefer the cascade heuristic.

While examining a nonlinear optimization problem of an economic model, Manne (1992) suggests discretizing the model into both a finite portion and an infinite portion. By adding a terminal condition to either the dual or primal variables, one can evaluate how the end effects change when the length of the finite portion is varied.

Furthermore, Manne found that choosing unequal time intervals vice equal ones over which to implement a rolling horizon heuristic minimized the approximation errors of a discount factor utility curve, though only by approximately 2%. This thesis conducts no research regarding unequal time intervals; however, Workman (2009) uses this technique when solving the Security Force Generation Model (SFGM), a manpower model that plans the force growth of both officers and enlisted soldiers of the Afghan National Army. SFGM is an infinite-horizon model that returns both monthly and annual goals. The objective function of SFGM includes discount penalties to incentivize decision-making early in the planning horizon. Workman chooses distinct windows in different phases of SFGM to allow for detail that is more precise where needed in the first part of the model and eliminate unnecessary detail to improve solve time in the latter part of the model. Specifically, SFGM solves in monthly increments for the first three years, and annually thereafter.

In the wake of extended delays to deliver the F-35 Lightning II Joint Strike Fighter to the United States Marine Corps (USMC) to replace the aging F/A-18 Hornet, Zerr (2016) develops the Hornet Assignment Sundown Model (HASM). HASM "prescribe[s] each individual Hornet's monthly squadron assignment, utilization, maintenance, storage, and retirement over its remaining service life while ensuring each squadron satisfies, to the extent possible, monthly flight hour requirements" (Zerr, 2016, p. 18). HASM solves by dividing its optimization monolith into 59 windows of six months each. The windows overlap by three months, and each iteration takes approximately 10–15 minutes to solve. Zerr (2016) used a trial-and-error technique to choose the lengths of six and three months for the window and advance, respectively. He selected these lengths to best satisfy his goal of both minimizing model solve time while still preventing myopic behavior.

Baker (1997) suggests that the length of each window and the advance between windows can have a significant effect on the quality of the solution produced using a cascade heuristic. Analysis of the results obtained throughout his research lead to the suggestion that the advance between windows should be, "at least as large as the

maximum number of time periods indices that are common to consecutively indexed rows" (Baker, 1997, p. 6). Additionally, Mercenier and Michel (1994) discuss the selection of the length of the window in conjunction with reformulating the infinite-time problem as a discrete-time approximation. The authors claim it is critical to ensure solutions of the discrete-time approximation are robust with respect to changes in the length of the window.

B. PREVIOUS WORK IN CORRECTING "END EFFECTS"

Selecting the length of a finite horizon approximation to an infinite-horizon problem must trade off computational speed with loss of optimality. The general belief is that as the length of the window increases, both the quality of solution obtained and the computation cost of solving the model increase.

While there is no formal definition of "end effects," the term generally describes the premature utilization of resources in an early time period or unrealistic prescriptions in later time periods caused by no visible future. An infinite planning horizon, especially with resources that regenerate over time, is challenging to represent in finite time with any fidelity to infinite-horizon consequences of near-term finite horizon decisions. As the length of the window decreases, the end effects of the model typically increase (Zerr, 2016).

There are four primary methods for correcting end effects: primal equilibrium approximation, dual equilibrium approximation, truncation, and salvage. While each method can be vitally important to improving solution quality, each technique also has significant shortfalls. Grinold (1983) provides both a quantitative and qualitative assessment of the strengths and weaknesses of these techniques and suggests model types for which each technique is best suited.

To achieve a "primal equilibrium approximation" to the monolith, Grinold adds additional constraints to restrict the model, which results in an upper bound of the optimal solution. For example, constraints are added to restrict all decision variables after a given month to the same value in an optimization model with monthly production decision variables. One shortcoming of this approximation is that a time period must

exist, “where a functional relationship can be derived that restricts the feasible region and leads to a finite-horizon re-formulation” (Walker, 1995, p. 22). Given a primal equilibrium approximation is a restriction, the approximation could be infeasible when the monolith is feasible. Conversely, if the monolith is infeasible any restriction is also infeasible.

To produce a lower bound to the optimal solution of a monolith, Grinold (1983) suggests a “dual equilibrium approximation.” This technique forms a relaxation of the monolith by aggregating constraints of the monolith after a given time period. This thesis similarly aggregates constraints after a given time period to produce a lower bound to our cascades.

The third technique for mitigating end effects, truncation, separates the model into time epochs and ignores the latter ones while solving the prior. While this is perhaps the easiest technique to implement, it can also result in unbounded solutions. To mitigate this, Grinold links epochs by adding a Lagrangian penalty to resources carried over from early ones to later ones, and refers to this as the salvage technique.

Workman (2009) compares the rate of convergence to an equilibrium approximation of the objective function values of the primal and dual SFGM. He structures SFGM to “approximate the infinite horizon that occurs at the end of the finite planning horizon” (Workman, 2009, p. 35). Additionally, Workman compares the convergence rate of a time truncation of the model designed to induce myopia by disregarding all events that occur outside its finite horizon. By comparing the results of the rate of convergence to equilibrium of the three models, Workman (2009) chooses the model best suited for his choice of periods. Zerr (2016) adds monthly penalties to reduce the effects of the myopic behavior induced by a six-month time window.

C. PREVIOUS WORK CONCERNING CONVERGENCE OF PRIMAL AND DUAL EQUILIBRIUMS

Baker (1997) develops a method for bounding the error resulting from a cascade heuristic for linear programs. Baker solves what is referred to as the “proximal cascade” with “a rolling-horizon technique to sequentially solve overlapping subsets of a SLP

[staircase linear program], where each subset is defined by a contiguous portion of the staircase” (Baker, 1997, p. 1). The bound for the error of the proximal cascade is then, “produced by a Lagrangian cascade, which solves sub problems that are also defined by contiguous portions of a staircase linear program (SLP), but are made separable by relaxing rows that would otherwise link columns from different sub problems” (Baker, 1997, p. 1). The Lagrangian cascade incorporates Lagrangian penalties, derived from the dual variables stored from previously solved sub-problems, in the objective function of the present sub-problem. Comparing the two cascade approximations provides a quantitative assessment of the accuracy of the proximal approximation solution. Baker found an average gap of 2.7% between the cascade solution and the monolith’s optimal solution when he applied this method to ten test problems. Furthermore, Baker attributes 60% of the gap reduction to the Lagrangian cascade. While this method works well for the test problems chosen, this is a very limited set of test problems and we know of no other research to guide implementation for more general problems.

III. MODEL FORMULATION

This thesis looks at two separate applications and their integer linear program (ILP) implementations, a production model (PM) and the revised Hornet Assignment Sundown Model (HASMa). We use these applications both when analyzing the impact on solution quality of varying the implementation of a cascade heuristic and when developing lower bounds for the optimal objective function value. This thesis also considers two variations of each model to produce lower bounds of the optimal objective function value. Production-1A and HASM-1A aggregate all periods following a designated time period. Production-2A and HASM-2A aggregate all time periods both before and after designated time periods.

A. PRODUCTION MODEL

Brown and Dell (2016) develop PM to serve as a simple test model for analyzing a cascade heuristic. This section presents the sets, data, decision variables, objective function value, and constraints that comprise the model.

1. Model Formulation

a. Indexed Sets [cardinality]

$f \in F$	facility [4]
$p \in P$	product [4]
$s \in S$	state (i.e. closed or open) alias s' [2]
$t \in T$	time period in planning horizon [12]

b. Data [units]

$demand_{p,t}$	units of demand for product *p* during time *t* [cases]
$end\_stor_{f,p}$	units of product *p* to be stored at facility *f* at end of planning horizon [cases]

$init\_state_f$	initial state of facility f [open or closed]
$\overline{make}_{f,p,t}$	maximum production by facility f of product p during time t [cases]
$start\_stor_{f,p}$	units of product p stored at facility f at start of planning horizon [cases]
$tran\_cost_{f,s,s'}$	transition cost for facility f from state s to s' [dollars]
$v\_cost_{f,p}$	variable cost per unit at facility f for product p [dollars]

c. Variables [units]

$MAKE_{f,p,t}$	units of product p made by facility f during time t [cases]
$SHIP_{f,p,t}$	units of product p shipped by facility f during time t [cases]
$STOR_{f,p,t}$	units of product p stored at facility f at end of time t [cases]
$TRAN_{f,s,s',t}$	=1 facility f transitions from state s to s' at end of time t [binary 0, or 1]

d. Formulation

$$\text{MIN}_{\substack{TRAN,PROD,\\ STOR,SHIP}} \sum_{f,s,s',t} tran\_cost_{f,s,s'} TRAN_{f,s,s',t} + \sum_{f,p,t} v\_cost_{f,p} MAKE_{f,p,t} \qquad (0)$$

subject to:

$$start\_stor_{f,p}\,|_{|t|=1} + STOR_{f,p,t-1}\,|_{|t|>1} + MAKE_{f,p,t} - STOR_{f,p,t} - SHIP_{f,p,t} = 0 \qquad \forall f \in F, p \in P, t \in T \qquad (1)$$

$$\sum_{f \in F} SHIP_{f,p,t} - demand_{p,t} = 0 \qquad \forall p \in P, t \in T \qquad (2)$$

$$-MAKE_{f,p,t} + \overline{make}_{f,p,t} \sum_{s \in S} TRAN_{f,s,'open',t} \geq 0 \qquad \forall f \in F, p \in P, t \in T \qquad (3)$$

$$\sum_{s \in S, s' \in S} TRAN_{f,s,s',t} - 1 = 0 \qquad \forall f \in F, \text{t} \in T \qquad (4)$$

$$1|_{|t|=1 \wedge s=init\_state_f} + \sum_{s' \in S} TRAN_{f,s',s,t-1} |_{|t|>1} - \sum_{s' \in S} TRAN_{f,s,s',t} = 0 \qquad \forall f \in F, s \in S, t \in T \qquad (5)$$

$$TRAN_{f,s,s',t} \in \{0,1\} \qquad \forall f \in F, p \in P, t \in T \qquad (6)$$

$$MAKE_{f,p,t} \in [0, \overline{make}_{f,p,t}] \qquad \forall f \in F, s \in S, s' \in S, t \in T$$

$$SHIP_{f,p,t} \in [0, demand_{p,t}] \qquad \forall f \in F, p \in P, t \in T$$

$$STOR_{f,p,t} \in [0, ubs_{f,p,t}] \qquad \forall f \in F, p \in P, t \in T$$

$$ubs_{f,p,t} = \min\{A_{f,p,t}, B_{p,t}, C_{f,p,t}\} \qquad \forall f \in F, p \in P, t \in T$$

e. Auxiliary Variables [units]

$$A_{f,p,t} = start\_stor_{f,p} + \sum_{t' \le t} \overline{make}_{f,p,t'} \text{ [cases]}$$

$$B_{p,t} = \max\left\{0, B_{p,t-1} + \sum_{f} \overline{make}_{f,p,t} - demand_{p,t}\right\} \text{ [cases]}$$

$$B_{p,\text{"0"}} = \sum_{f} start\_stor_{f,p} \text{ [cases]}$$

$$C_{f,p,t} = \sum_{t'>t} demand_{p,t'} + end\_stor_{f,p} \text{ [cases]}$$

2. Explanation of Model Formulation

Equation (0) is the objective function of the model. There are two components to the function:

(a) $\sum_{f,s,s',t} tran\_cost_{f,s,s'} TRAN_{f,s,s',t}$, and

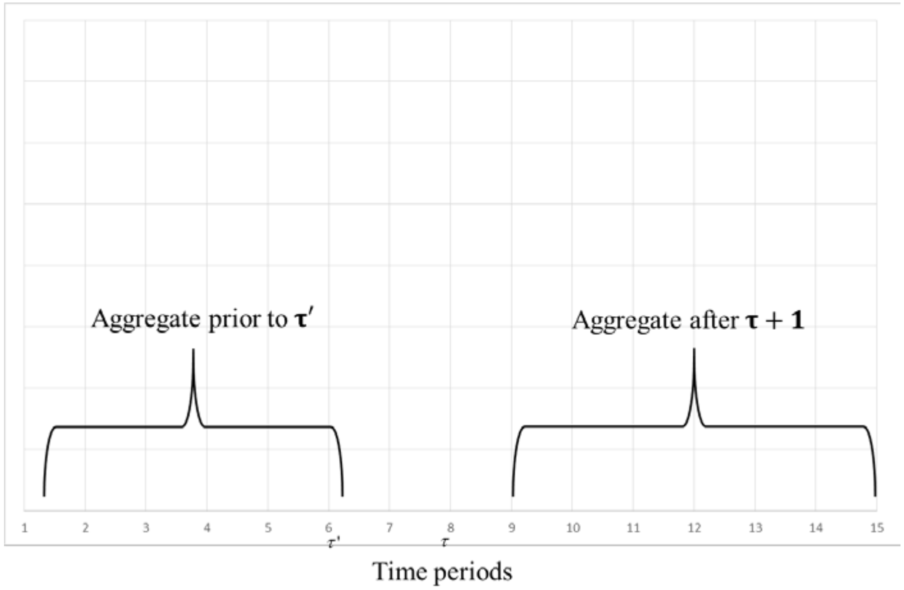
(b) $\sum_{f,p,t} v\_cost_{f,p} MAKE_{f,p,t}$.

Component (a) expresses the cost of opening or closing each facility during each period. Component (b) expresses the cost of making each product at each facility during each period. Each constraint (1) ensures the initial units of a product at a facility and the

quantity of some product stored or made at that facility at the end of the previous period is equivalent to the quantity of that product shipped or stored at that facility at the end of the period. Each constraint (2) ensures the amount of a product shipped from a facility is equivalent to the demand for that product during a period. Each constraint (3) ensures that the quantity of a product made at a facility does not exceed the maximum production capability, for a facility that is open. Each constraint (4) ensures that a facility makes no more than one transition during a period. Each constraint (5) keeps track of the state of a facility and ensures only proper transitions occur. Constraints (6) define domains for the variables. In addition, some auxiliary variables are computed from model values. $A_{f,p,t}$ is the maximum amount of each product *p* that can exist up to time *t* for facility *f*. $B_{p,t}$ is the maximum amount of storage possible based on the total product limits. $B_{p,"0"}$ is the total initial amount of each product. $C_{f,p,t}$ is the maximum amount of each product, at each time period and each facility, that would ever be required in the future.

B. PRODUCTION MODEL AGGREGATION

An ILP with aggregated constraints for periods 1 to τ' and $\tau+1$ to T^{MAX}, as shown in Figure 2, produces a lower bound on the optimal objective function value because it is a relaxation of the monolith. In the aggregated segments, additional constraints sum all time periods together. The aggregated PM breaks the periods of the model into three segments: an aggregation of a contiguous set of beginning periods, a contiguous segment when time periods are not aggregated, and a following aggregation to the end of the model. The aggregation of PM maintains all the same sets, data, variables, formulation, and constraints as the original production model. However, the PM aggregation adds two new parameters, τ and τ', where τ indicates the last period of the non-aggregated segment and τ' indicates the last period of the former aggregation. This thesis consider two variations of the aggregated PM. Production-1A fixes $\tau'=1$ for all iterations while $\tau=1,2,...,11$. Production-2A considers values of τ and τ' where $\tau=2,3,...,12$ and $\tau'=1,2,...,11$.



In this model, $\tau = 8$ and $\tau' = 6$. The beginning aggregation consists of periods 1–5, and the later aggregation consists of periods 9–15. The model does not aggregate periods 6–8 and these periods solve under the same conditions as the monolith.

Figure 2. Example of an Aggregated Model

1. Aggregated Production Model Modifications

$\tau + 1$ First period of later aggregation

τ' Last period of early aggregation

$$start\_stor_{f,p}\mid_{|t|=1} + STOR_{f,p,t\text{-}1}\mid_{|t|>1} + MAKE_{f,p,t} - STOR_{f,p,t} - SHIP_{f,p,t} = 0 \qquad \forall f \in F, p \in P, \; t \mid \tau' < t \le \tau \qquad (1)$$

$$\sum_{t=\tau+1}^{T^{MAX}} \begin{pmatrix} start\_stor_{f,p}\mid_{|t|=1} + STOR_{f,p,t\text{-}1}\mid_{|t|>1} + MAKE_{f,p,t} \\ -STOR_{f,p,t} - SHIP_{f,p,t} \end{pmatrix} = 0 \qquad \forall f \in F, p \in P \qquad (1a)$$

$$\sum_{t=1}^{\tau'} \begin{pmatrix} start\_stor_{f,p}\mid_{|t|=1} + STOR_{f,p,t\text{-}1}\mid_{|t|>1} + MAKE_{f,p,t} \\ -STOR_{f,p,t} - SHIP_{f,p,t} \end{pmatrix} = 0 \qquad \forall f \in F, p \in P \qquad (1b)$$

$$\sum_{f \in F} SHIP_{f,p,t} - demand_{p,t} = 0 \qquad \forall p \in P, \; t \mid \tau' < t \le \tau \qquad (2)$$

$$\sum_{t=\tau+1}^{T^{MAX}} \left(\sum_{f \in F} SHIP_{f,p,t} - demand_{p,t} \right) = 0 \qquad \forall p \in P \qquad (2a)$$

$$\sum_{t=1}^{\tau'} \left(\sum_{f \in F} SHIP_{f,p,t} - demand_{p,t} \right) = 0 \qquad \forall p \in P \qquad (2b)$$

$$-MAKE_{f,p,t} + \overline{make}_{f,p,t} \sum_{s \in S} TRAN_{f,s,'open',t} \geq 0 \qquad \forall f \in F, p \in P, \; t \mid \tau' < t \leq \tau \qquad (3)$$

$$\sum_{t=\tau+1}^{T^{MAX}} \left(-MAKE_{f,p,t} + \overline{make}_{f,p,t} \sum_{s \in S} TRAN_{f,s,'open',t} \right) \geq 0 \qquad \forall f \in F, p \in P \qquad (3a)$$

$$\sum_{t=1}^{\tau'} \left(-MAKE_{f,p,t} + \overline{make}_{f,p,t} \sum_{s \in S} TRAN_{f,s,'open',t} \right) \geq 0 \qquad \forall f \in F, p \in P \qquad (3b)$$

$$\sum_{s \in S, s' \in S} TRAN_{f,s,s',t} - 1 = 0 \qquad \forall f \in F, \; t \mid \tau' < t \leq \tau \qquad (4)$$

$$\sum_{t=\tau+1}^{T^{MAX}} \left(\sum_{s \in S, s' \in S} TRAN_{f,s,s',t} - 1 \right) = 0 \qquad \forall f \in F \qquad (4a)$$

$$\sum_{t=1}^{\tau'} \left(\sum_{s \in S, s' \in S} TRAN_{f,s,s',t} - 1 \right) = 0 \qquad \forall f \in F \qquad (4b)$$

$$\begin{aligned} 1\big|_{|t|=1 \wedge s=init\_state_f} &+ \sum_{s' \in S} TRAN_{f,s',s,t-1}\big|_{|t|>1} \\ &- \sum_{s' \in S} TRAN_{f,s,s',t} = 0 \end{aligned} \qquad \forall f \in F, s \in \mathrm{S}, \; t \mid \tau' < t \leq \tau \qquad (5)$$

$$\sum_{t=\tau+1}^{T^{MAX}} \left(\begin{aligned} 1\big|_{|t|=1 \wedge s=init\_state_f} &+ \sum_{s' \in S} TRAN_{f,s',s,t-1}\big|_{|t|>1} \\ &- \sum_{s' \in S} TRAN_{f,s,s',t} \end{aligned} \right) = 0 \qquad \forall f \in F, s \in \mathrm{S} \qquad (5a)$$

$$\sum_{t=1}^{\tau'} \left(\begin{aligned} 1\big|_{|t|=1 \wedge s=init\_state_f} &+ \sum_{s' \in S} TRAN_{f,s',s,t-1}\big|_{|t|>1} \\ &- \sum_{s' \in S} TRAN_{f,s,s',t} \end{aligned} \right) = 0 \qquad \forall f \in F, s \in \mathrm{S} \qquad (5b)$$

2. Explanation of Model Formulation

Each of the original constraints of PM are modified so that they are only considered for time periods between $\tau'+1$ and τ. The aggregated PM adds two additional constraint sets for each of the original constraint sets (1) through (5). One set aggregates the periods after τ, while the other aggregates the periods before τ'.

C. HORNET ASSIGNMENT SUNDOWN MODEL

1. Background

Zerr (2016) formulates and implements the ILP HASM to optimally assign aircraft from the USMC F/A-18 Hornet inventory to various squadrons and maintenance depot locations to satisfy the operational flight hour requirements of Hornet squadrons, while simultaneously meeting aircraft maintenance requirements. Production delays of the F-35 Lightning II Joint Strike Fighter, the replacement aircraft for the F/A-18, have forced the Marine Corps to extend the service life of the F/A-18 beyond its initial capability, creating the concern that F/A-18 flight hour requirements could be satisfied while still performing both routine required maintenance and service life extensions (Zerr, 2016). HASM assigns each aircraft to an optimal location during each month of the next 15 years to minimize the cost of required depot maintenance, while still maintaining the number of required squadron flying hours as they slowly transition to the F-35. Zerr's (2016) version of HASM considers transitions of 274 aircraft to and from 13 squadrons over 176 months and to five different depot maintenance events. This model consists of "more than 28.8 million rows, 80 million columns (77 million discrete columns), and 388 million non-zero elements," (Zerr, 2016, p. 37) and is far too large to solve as a monolith. As a result, the author solves HASM using a cascade heuristic with a window of six months and an advance of three months.

We simplify HASM to reduce run time and make the monolith solvable and refer to the revised version of the model as HASMa. Having a solvable monolith provides a baseline to compare the quality of solution produced using a cascade heuristic. The simplifications of HASMa include aggregating the time periods from months into years, considering only three depot events vice five, and aggregating the 13 flying squadrons

into one squadron that indicates utilization of the aircraft in the operating forces. HASMa keeps the three primary depot events, per Zerr (2016): planned maintenance interval (PMI), initial service life extension (SLE), and high flight hour (HFH) extension. SLE increases the service life of an aircraft from 6,000 to 8,000 flight hours, and the HFH upgrade adds an additional 2,000 flight hours to the service life. Depots cannot conduct HFH maintenance until the initial SLE is completed. The two depot events eliminated in HASMa are combinations of PMI with either SLE or HFH, events which Zerr found had little significance in increasing aircraft availability. Zerr gathered the original inputs to HASM from four unclassified data sources: the 2016 USMC Aviation Plan from Headquarters Marine Corps Aviation, flight hour inventory reports from Naval Air Systems Command (NAVAIR), Aviation Maintenance/Supply Readiness Report Data from Headquarters Marine Corps Aviation, and Depot Maintenance Data from NAVAIR. This thesis added no additional or supplemental data to HASMa.

2. Model Formulation

This section presents the indexed sets, data, variables, formulation, and constraints that comprise the formulation of HASMa, modified from the original composition of HASM by Zerr (2016).

a. Indexed Sets [cardinality]

A — The set of all F/A-18 Hornet aircraft [274]

$a \in A = \{1...N\}$

ALLOW — The set of allowable locations at time *t* [105]

$(l,t) \in ALLOW$

DEP — The set of all depot maintenance activities [3]

$l \in DEP = \{PMI, SLE, HFH\}$

HFHC — The set of initial aircraft that have received HFH [111]

$a \in HFHC$

INIT — The set of *a* which begin HASMa at *depot* [115]

$(a,l) \in INIT$

L The set of all non-squadron locations [6]

$l \in L$, $L = \{PMI, SLE, HFH, Store, Backlog, Retire\}$

$SLEC$ The set of initial aircraft that have received SLE [247]

$a \in SLEC$

T The set of all time periods [years] [15]

$t \in T = \{1 \ldots T^{MAX}\}$

b. Data [units]

Initial Conditions [units]

$fT0_a$ The total flying hours for aircraft *a* at *t*=1. [flight hours]

$jT0_a$ The time period that aircraft *a* must depart the depot if located at the depot at the beginning of model. [year]

$PMI0_a$ The next PMI for aircraft *a* must be completed on or before this year. [year]

$xT0_{a,l}$ The initial starting condition for aircraft *a* at *t*=1. [indicator 0, or 1]

Depot Data [units]

cap Capacity restriction on aircraft at depot locations. [aircraft]

$dmax_a$ The maximum number of years between PMI events for aircraft *a*. [years]

$dmin_a$ The minimum number of years between PMI events for aircraft *a*. [years]

$hhours_t$ Number of hours of HFH maintenance in the t^{th} year of maintenance. [man-hours]

$Lwin\_1_a$ Lower limit for 1st PMI event for aircraft *a*, similarly $Lwin\_2_a, Lwin\_3_a, Lwin\_4_a$ are the lower limits for 2nd, 3rd, and 4th PMI events. [years]

$phours_t$ Number of hours of PMI maintenance in the t^{th} year of maintenance. [man-hours]

q^{ot}	Number of overtime depot maintenance hours. [man-hours]
q^{reg}	Number of regular depot maintenance hours. [man-hours]
$shours_t$	Number of hours of SLE maintenance in the t^{th} year of maintenance. [man-hours]
$time_l$	Number of years to complete depot event at location *l*. [years]
u_t^{ot}	Cost of an hour of overtime depot level work for any event. [penalty/hour]
$Uwin\_1_a$	Upper limit for 1st PMI event for aircraft *a*, similarly $Uwin\_2_a, Uwin\_3_a, Uwin\_4_a$ are the upper limits for 2nd, 3rd, and 4th PMI events. [years]

Flight Hour and Readiness [units]

$\underline{h}_t$	Minimum cumulative flying hours for all aircraft in year *t*. [flight hours]
$\overline{h}_t$	Maximum flying hours for a single aircraft in year *t*. [flight hours]
m	Minimum percentage of $\underline{h}_t$ that must be completed by each aircraft not in a depot location in *t*. [flight hours]
r_t	Ready Basic Aircraft (RBA) rate at *t*. [fraction of RBA]

Aircraft Assignment [units]

$\overline{n}$	Maximum number of serviceable aircraft at Squadron location. [aircraft]
$\underline{n}$	Minimum number of serviceable aircraft at Squadron location. [aircraft]

Penalties [units]

Elastic penalties	Elastic penalties for violating constraint equations
$w_{l,t}^{pen}$	Penalty per aircraft per year for non-squadron location. [penalty/aircraft]
$w_{l,l'}^{xfr}$	Penalty per aircraft to transfer as aircraft from l to l' in time *t*. [penalty/aircraft]
$pens_{a,l'}$	Penalty multiplier for aircraft *a* for preference of transfer location. [penalty/penalty]

c. *Binary Variables*

$H_{a,t} \in \{0,1\}$ Binary variable with value of one if aircraft *a* has completed *HFH* on or before time *t*, zero otherwise.

$R_{a,t} \in \{0,1\}$ Binary variable with value of one if aircraft a retires on or before time t, zero otherwise.

$S_{a,t} \in \{0,1\}$ Binary variable with value of one if aircraft *a* has completed *SLE* on or before time *t*, zero otherwise.

$X_{a,l,t} \in \{0,1\}$ Binary variable with value of one if aircraft *a* is in location *l* at the start of year *t*, zero otherwise.

$Y_{a,l,l',t} \in \{0,1\}$ Binary variable with value of one if aircraft *a* transfers out of location *l* into location l' at the start of the year *t*, zero otherwise.

d. *Nonnegative Variables [units]*

$F_{a,t}$ Number of flight hours assigned to aircraft *a* during year *t*. [hours]

$F_{a,t}^{tot}$ Cumulative flight hours assigned to aircraft *a* up to and including year *t*. [hours]

V_t^{ot} Number overtime hours used at depot during year *t*. [hours]

e. *Formulation*

$$\text{MIN} \sum_a \left(\sum_{l,t \in ALLOW} w_{l.t}^{pen} X_{a,l,t} + \sum_{l,l',t \in TRANS} w_{l,l'}^{xfr}\, pens_{a,l'} Y_{a,l,l',t} \right) + \sum_t u_t^{ot} V_t^{ot} + Elastic\ \ penalties \tag{0}$$

Subject to:

$$\sum_a F_{a,t} \le r_t \bar{h}_t \sum_{a,l \notin L} X_{a,l,t} \qquad \forall t \in T \tag{1}$$

$$F_{a,t} \le \bar{h}_t \sum_{l \notin L} X_{a,l,t} \qquad \forall a \in A, t \in T \tag{2}$$

$$F_{a,t} \ge m\underline{h}_t \sum_{l \notin L} X_{a,l,t} \qquad \forall a \in A, t \in T \tag{3}$$

$$\sum_a F_{a,t} \dot{\geq} \underline{h}_t \qquad \forall t \in T \qquad (4)$$

$$F_{a,t}^{tot} = F_{a,t} + fT0_a \qquad \forall a \in A, t = 1 \qquad (5)$$

$$F_{a,t}^{tot} = F_{a,t} + F_{a,t-1}^{tot} \qquad \forall a, t > 1 \qquad (6)$$

$$F_{a,t}^{tot} \leq 10000 \qquad \forall a \in HFHC, t \in T \qquad (7)$$

$$F_{a,t}^{tot} \leq 8000 + 2000 H_{a,t} \qquad \forall a \in SLEC, t \in T \qquad (8)$$

$$F_{a,t}^{tot} \leq 6000 + 2000 H_{a,t} + 2000 S_{a,t} \qquad \forall a \notin SLEC \cup HFHC, t \in T \qquad (9)$$

$$X_{a,l,t} = xT0_{a,l} \qquad \forall a \in A, l \in L, t = 1 \qquad (10)$$

$$\sum_{l|(l,t) \in ALLOW} X_{a,l,t} = 1 \qquad \forall a \in A, t \in T \qquad (11)$$

$$X_{a,l,t} = X_{a,l,t-1} + \sum_{l'} Y_{a,l',l,t-1} - \sum_{l'} Y_{a,l,l',t} \qquad \forall a \in A, l \in L, t > 1 \qquad (12)$$

$$\sum_{a,l \notin L} X_{a,l,t} \dot{\leq} \overline{n} \qquad \forall t \in T \qquad (13)$$

$$\sum_{a,l \notin L} X_{a,l,t} \dot{\geq} \overline{n} \qquad \forall t \in T \qquad (14)$$

$$\sum_{a,l \in DEP} X_{a,l,t} \leq cap \qquad \forall t \in T \qquad (15)$$

$$X_{a,l,t} \geq \sum_{l'} Y_{a,l,l',t} \qquad \forall a \in A, (l,t) \in ALLOW \qquad (16)$$

$$\sum_{l'} Y_{a,l',l,t} = \sum_{l'} Y_{a,l,l',t'} \qquad \forall a \in A, l \in DEP, t \in T, t' = t + time_l \qquad (17)$$

$$\sum_{l'} Y_{a,l,l',t} = 1 \qquad (a,l) \in INIT, t = jT0_a \qquad (18)$$

$$\sum_{l,t'|t' \leq t} Y_{a,l,PMI,t'} + X_{a,Store,t} + X_{a,Backlog,t} + X_{a,Retire,t} + X_{a,SLE,t} + X_{a,HFH,t} \geq 1 \qquad \forall a \in A, t | 1 < PMI0_a \leq t \qquad (19)$$

$$\sum_{t-dmax_a < t' \leq t} X_{a,PMI,t'} + X_{a,Store,t} + X_{a,Retire,t} + X_{a,SLE,t} + X_{a,HFH,t} + X_{a,Backlog,t} + \sum_l Y_{a,l,PMI,t} \geq 1 \qquad \forall a \in A, t = dmax_a \qquad (20)$$

$$\sum_{l,t'|t-dmin_a < t' \leq t} Y_{a,l,PMI,t'} \leq 1 \qquad \forall a \in A \qquad (21)$$

$$H_{a,t} \leq \sum_{l,t'|t' \leq t-time_{HFH}} Y_{a,l,HFH,t'} \qquad \forall a \notin HFHC, t \in T \qquad (22)$$

$$S_{a,t} \leq \sum_{l,t'|t' \leq t-time_{SLE}} Y_{a,l,SLE,t'} \qquad \forall a \notin SLEC, t \in T \qquad (23)$$

$$\sum_{a,l,t'|t-time_{HFH}+1 \leq t'} hhours_{t-t'+1} Y_{a,l,HFH,t'} + \sum_{a,l,t'|t-time_{PMI}+1 \leq t'} phours_{t-t'+1} Y_{a,l,PMI,t'} + \sum_{a,l,t'|t-time_{SLE}+1 \leq t'} shours_{t-t'+1} Y_{a,l,SLE,t'} \leq q^{reg} + V_t^{ot} \qquad \forall t \in T \qquad (24)$$

$$V_t^{ot} \leq q^{ot} \qquad \forall t \in T \qquad (25)$$

$$R_{a,t} \leq \sum_{l,t'} Y_{a,l,Retire,t'} \qquad \forall a \in A \qquad (26)$$

$$\sum_{l,t \; |Lwin\_1_a \leq t \leq Uwin\_1_a} Y_{a,l,PMI,t} \leq 1 \qquad \forall a \in A \,|\, Lwin\_1_a \leq T^{MAX} \qquad (27a)$$

$$\sum_{l,t \; |Lwin\_2_a \leq t \leq Uwin\_2_a} Y_{a,l,PMI,t} \leq 1 \qquad \forall a \in A \,|\, Lwin\_2_a \leq T^{MAX} \qquad (27b)$$

$$\sum_{l,t \; |Lwin\_3_a \leq t \leq Uwin\_3_a} Y_{a,l,PMI,t} \leq 1 \qquad \forall a \,|\, Lwin\_3_a \leq T^{max} \qquad (27c)$$

$$\sum_{l,t \; |Lwin\_4_a \leq t \leq Uwin\_4_a} Y_{a,l,PMI,t} \leq 1 \qquad \forall a \,|\, Lwin\_4_a \leq T^{max} \qquad (27d)$$

$$\sum_{l,t} Y_{a,l,HFH,t} \leq 1 \qquad \forall a \notin HFHC \qquad (28a)$$

$$\sum_{l,t} Y_{a,l,\mathrm{SLE},t} \leq 1 \qquad \forall a \notin SLEC \qquad (28b)$$

$$F_{a,t} \geq 0 \qquad \forall a \in A, t \in T \qquad (29)$$

$$F_{a,t}^{tot} \geq 0 \qquad \forall a \in A, t \in T \qquad (30)$$

$$V_t^{ot} \geq 0 \qquad \forall t \in T \qquad (31)$$

$$H_{a,t}; S_{a,t} \in \{0,1\} \qquad \forall a \in A, t \in T \qquad (32\text{–}33)$$

$$X_{a,l,t} \in \{0,1\} \qquad \forall a \in A, l \in L, t \in T \qquad (34)$$

$$Y_{a,l,l',t} \in \{0,1\} \qquad \forall a \in A, l \in L, l' \in \mathrm{L}\; t \in T \qquad (35)$$

3. Explanation of Model Formulation

Equation (0) is the objective function. To discourage violations of flight hour requirements and aircraft assignments, both piecewise linear penalties and a discount factor are included, as originally described by Zerr (2016). There are four components to the objective function:

(a) $\sum_{a,(l,t)\in ALLOW} w_{l,t}^{pen} X_{a,l,t}$,

(b) $\sum_{a,(l,l',t)\in TRANS} w_{l,l'}^{xfr} pens_{a,l'} Y_{a,l,l',t}$,

(c) $\sum_{t} u_t^{ot} V_t^{ot}$, and

(d) *Elastic penalties.*

Component (a) and component (b), per Zerr (2016), describe the penalties associated with transferring aircraft between locations and removing them from the operational squadron. Component (c) calculates the overtime costs incurred at the depot when conducting maintenance. Component (d) represents elastic penalties for violating constraints.

Each constraint (1) and (2), per Zerr (2016), ensures that the flight hours assigned to each aircraft remains below the number of hours an aircraft has the ability to fly, based on its service life restrictions. Each constraint (3) ensures every aircraft flies a minimum percentage of the total yearly flight hour goal. Each constraint (4) "balances the total number of flight hours assigned to a squadron with the minimum required and deviations below that amount" (Zerr, 2016, p. 35). Each constraint (5) and (6) tracks the total number of hours each aircraft has flown. Each constraint (7) through (9) imposes service life restrictions on each aircraft based on the maintenance it has completed.

The next set of constraints track each aircraft's location during each year. Each constraint (10), (11), and (12) "establish the initial position of each aircraft at the start of the model, limit an aircraft to one unique location during any time period, and connect each aircraft's location to where it was in the previous time period" (Zerr, 2016, p. 35). Each constraint (13) and (14) limits the number of aircraft assigned to the squadron "between a floor minimum number and an elastic maximum number" (Zerr, 2016, p. 35). Each constraint (15) limits the number of aircraft receiving maintenance.

Each constraint (16) through (21) tracks the flow of aircraft in and out of the three maintenance events and prevents excessive transferring of aircraft. Each constraint (16) ensures an aircraft is only transferred to a new location if it is allowed to be transferred to

that location during the given time period. Each constraint (17) forces aircraft to remain in maintenance for enough consecutive periods to complete the event. Each constraint (18) ensures aircraft assignment to one location each period. Each constraint (19) through (21) "require every aircraft to complete the first scheduled PMI event and every subsequent PMI event unless the aircraft is placed in storage or retired" (Zerr, 2016, p. 36). Each constraint (22) and (23) updates an aircraft's service life based on the maintenance it has completed. Each constraint (24) tracks the maintenance hours worked by depot employees, and each constraint (25) dictates the maximum number of overtime hours available each year. Each constraint (26) tracks aircraft that have been retired. Each constraint (27) ensures an aircraft does not receive PMI before it is required. Each constraint (28) prevents an aircraft from completing the same service life extension more than once. Each constraint (29) through (31) gives non-negative domains for variables, and (32) through (35) indicate the binary decision variables.

D. HASMa AGGREGATION

Similar to the aggregation of PM, in the aggregated version of HASMa, the periods of the model are broken into three epochs: an aggregation of the beginning periods, a segment of periods not aggregated, and an aggregation of the end of the model. The aggregated model maintains all the same sets, data, binary variables, nonnegative variables, formulation, and constraints as HASMa. However, the HASMa aggregation adds two new parameters, τ and τ'. The parameter τ' indicates the last period of the former aggregation while τ indicates the last period of the non-aggregated segment. For analyzing results, this thesis considers two variations of the HASMa aggregation. HASM-1A set $\tau' = 1$ for all iterations while $\tau = 1, 2, ..., 14,$ meaning that there was no aggregation in the beginning of the model. HASM-2A considers all possible combinations of τ and τ' where $\tau = 2, 3, ..., 14$ and $\tau' = 1, 2, ..., 13$, which results in 91 trial runs.

1. Aggregated HASMa Modifications

$\tau + 1$ First period of later aggregation

τ' Last period of early aggregation

$$\sum_{a} F_{a,t} \le r_t \bar{h}_t \sum_{a,l \notin L} X_{a,l,t} \qquad \forall t \mid \tau' < \mathrm{t} \le \tau \qquad (1)$$

$$\sum_{a,t=\tau+1}^{T^{MAX}} F_{a,t} \le \sum_{t=\tau+1}^{T^{MAX}} r_t \bar{h}_t \sum_{a,l \notin L} X_{a,l,t} \qquad (1a)$$

$$\sum_{a,t=1}^{\tau'} F_{a,t} \le \sum_{t=1}^{\tau'} r_t \bar{h}_t \sum_{a,l \notin L} X_{a,l,t} \qquad (1b)$$

$$F_{a,t} \le \bar{h}_t \sum_{l \notin L} X_{a,l,t} \qquad \forall a \in A, \ \forall t \mid \tau' < t \le \tau \qquad (2)$$

$$\sum_{t=\tau+1}^{T^{MAX}} F_{a,t} \le \sum_{t=\tau+1, l \notin L}^{T^{MAX}} \bar{h}_t X_{a,l,t} \qquad \forall a \in A \qquad (2a)$$

$$\sum_{t=1}^{\tau'} F_{a,t} \le \sum_{t=1, l \notin L}^{\tau'} \bar{h}_t X_{a,l,t} \qquad \forall a \in A \qquad (2b)$$

$$F_{a,t} \ge m \underline{h}_t \sum_{l \notin L} X_{a,l,t} \qquad \forall a \in A, \ \forall t \mid \tau' < t \le \tau \qquad (3)$$

$$\sum_{t=\tau+1}^{T^{MAX}} F_{a,t} \ge \sum_{t=\tau+1, l \notin L}^{T^{MAX}} m \underline{h}_t X_{a,l,t} \qquad \forall a \in A \qquad (3a)$$

$$\sum_{t=1}^{\tau'} F_{a,t} \ge \sum_{t=1, l \notin L}^{\tau'} m \underline{h}_t X_{a,l,t} \qquad \forall a \in A \qquad (3b)$$

$$\sum_{a} F_{a,t} \dot{\ge} \underline{h}_t \qquad \forall t \mid \tau' < \mathrm{t} \le \tau \qquad (4)$$

$$\sum_{a,t=\tau+1}^{T^{MAX}} F_{a,t} \dot{\ge} \sum_{t=\tau+1}^{T^{MAX}} \underline{h}_t \qquad (4a)$$

$$\sum_{a,t=1}^{\tau'} F_{a,t} \dot{\ge} \sum_{t=1}^{\tau'} \underline{h}_t \qquad (4b)$$

$$F_{a,t}^{tot} = F_{a,t} + F_{a,t-1}^{tot} \qquad \forall a \in A, \ \forall t \mid \tau' < t \le \tau \qquad (6)$$

$$F_{a,t=\tau+1}^{tot} = \sum_{t=\tau+1}^{T^{MAX}} \left(F_{a,t} \right) + F_{a,t=\tau}^{tot} \qquad \forall a \in A \qquad (6a)$$

$$F_{a,t=\tau+1}^{tot} = \sum_{t=1}^{\tau'} \left(F_{a,t} \right) \qquad \forall a \in A \qquad (6b)$$

$$F_{a,t}^{tot} \le 10000 \qquad \forall a \in HFHC, \ \forall t \mid \tau' < t \le \tau \qquad (7)$$

$$F_{a,t}^{tot} \le 10000 \qquad \forall a \in HFHC, t = \tau + 1 \qquad (7a)$$

$$F_{a,t}^{tot} \le 10000 \qquad \forall a \in HFHC, t = \tau' \qquad (7b)$$

$$F_{a,t}^{tot} \leq 8000 + 2000H_{a,t} \qquad \forall a \in SLEC, \; \forall t \mid \tau' < t \leq \tau \qquad (8)$$

$$F_{a,t=\tau+1}^{tot} \leq 8000 + 2000\sum_{t=\tau+1}^{T^{MAX}} H_{a,t} \qquad \forall a \in SLEC \qquad (8a)$$

$$F_{a,t=1}^{tot} \leq 8000 + 2000\sum_{t=1}^{\tau'} H_{a,t} \qquad \forall a \in SLEC \qquad (8b)$$

$$F_{a,t}^{tot} \leq 6000 + 2000H_{a,t} + 2000S_{a,t} \qquad \forall a \notin SLEC \cup HFHC, \; \forall t \mid \tau' < t \leq \tau \qquad (9)$$

$$F_{a,t=\tau+1}^{tot} \leq 6000 + 2000\sum_{t=\tau+1}^{T^{MAX}} H_{a,t} + 2000\sum_{t=\tau+1}^{T^{MAX}} S_{a,t} \qquad \forall a \notin SLEC \cup HFHC \qquad (9a)$$

$$F_{a,t=1}^{tot} \leq 6000 + 2000\sum_{t=1}^{\tau'} H_{a,t} + 2000\sum_{t=1}^{\tau'} S_{a,t} \qquad \forall a \notin SLEC \cup HFHC \qquad (9b)$$

2. Explanation of Model Formulation

All constraints from HASMa not listed in the modifications given previously also appear unchanged in the HASMa aggregation. The aggregated model considers aggregated constraints for HASMa constraints (1) through (4) and (6) through (9). HASMa aggregates these constraints because they are responsible for assigning the proper number of flight hours to each available aircraft, ensuring assignment of the minimum flight hours to the squadron, and ensuring each aircraft flies only the number of hours that its service life limitation allows. It was not necessary to aggregate constraints (10) through (40) of HASMa as they primarily consider the logistics of tracking and updating an aircraft's location. The aggregated model also does not include an aggregated constraint for constraint (5) of HASMa, as this constraint considers the initial location of the aircraft and only effects the first period. Wherever the aggregated model adds a constraint, it also modifies the original constraint. The original constraint is now only applicable for periods $\tau'+1$ to τ .

THIS PAGE INTENTIONALLY LEFT BLANK

IV. IMPLEMENTATION AND ANALYSIS

This chapter provides insight into the implementation and analysis of the models considered. This thesis uses the Generalized Algebraic Modeling System (GAMS) version 24.6.1 to generate all models and CPLEX 12.0 to solve them. (GAMS, n.d.). We solved all models on a Dell computer using two 2.30GHz processors and 128 GB RAM.

We solve the monolithic forms of both PM and HASMa prior to solving the model with the cascade heuristic. The objective function values reported for the monoliths are the true optimal solution for each respective problem. In the following, we refer to the optimal minimization solution as the monolith solution. These solutions provide a baseline for comparison of all other objective function values reported with the cascade heuristic and with the model aggregations.

A. PRODUCTION MODEL CASCADE IMPLEMENTATION

This thesis solves PM with 55 different cascades, with the window and advance varied for each run. With only 1,729 decision variables and 577 constraints, the monolith solves in less than a second. In the runs considered, the window takes on values 2, 3... 11, while the advance takes on values of 1, 2... 10 respectively. The trial runs consider every possible combination of window and advance between these respective values. Additionally, each run of the model requires a 0.0% optimality gap (a guaranteed optimal solution). PM generates its data using the formulas provided below with the given values.

$$demand\_scale = 1000 \tag{1}$$

$$facility\_scale = 1000 \tag{2}$$

$$make\_scale = 3 \tag{3}$$

$$product\_scale = 10 \tag{4}$$

$$tran\_scale = \tag{5}$$

State From	State To	Cost
closed	closed	0.1
closed	open	1.0
open	open	0.4
open	closed	2.0

$$demand_{p,t} = \max\{0, D\}$$
$$\text{where } D \sim \mathrm{N}\left(demand\_scale * seasonal_t, 0.5 * demand\_scale * seasonal_t\right) \tag{6}$$

$$end\_stor_{f,p} = 0 \qquad \forall f \in F, p \in P \tag{7}$$

$$init\_state_f = \tag{8}$$

Facility	***init_state***
1	open
2	open
3	closed
4	closed

$$\overline{make}_{f,p,t} = make\_scale * \frac{\sum_t demand_{p,t}}{t-1} \tag{9}$$

$$seasonal_t =$$

time period	1	2	3	4	5	6	7	8	9	10	11	12
Seasonality	1.0	1.0	1.0	1.5	1.0	1.0	1.0	1.0	1.0	1.1	2.0	2.5

(10)

$$start\_stor_{f,p} = 0 \qquad \forall f \in F, p \in P \tag{11}$$

$$tran\_cost_{f,s,s'} = \max\{0, TC\} \tag{12}$$
$$\text{where } TC \sim \mathrm{N}\begin{pmatrix} tran\_scale_{s,s'} * facility\_scale, \\ 0.1 * tran\_scale_{s,s'} * facility\_scale \end{pmatrix}$$

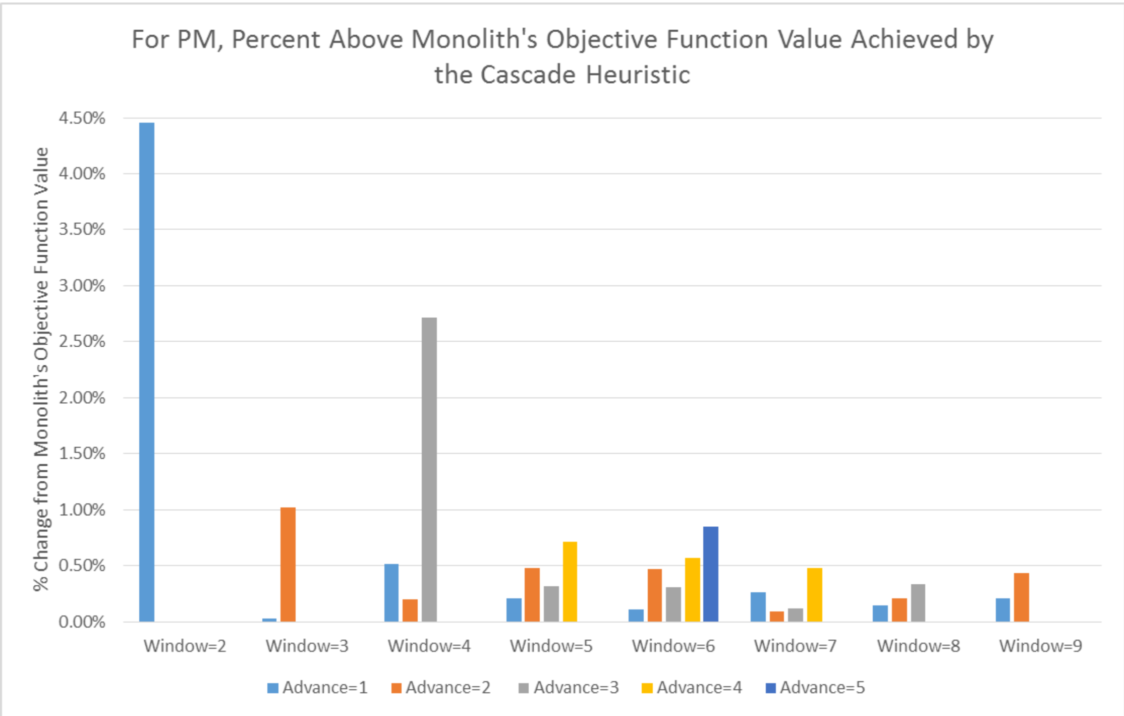
$$v\_cost_{f,p} = \max\{1, \mathrm{VC}\} \tag{13}$$
$$\text{where } \mathrm{VC} \sim \mathrm{N}\left(product\_scale, 0.5 * product\_scale\right)$$

Table 2 shows the percent increase from the monolith's objective function value for each of the 55 runs of the cascade of PM. For example, with an advance of one and a window of two, the cascade heuristic solution is 4.45% above the monolith's objective function value. The quality of the solutions from each run is very good. Figure 3 highlights that holding the length of the window constant while simultaneously increasing the length of the advance results in neither a monotonically increasing nor decreasing change in the objective function value. With the exception when window=7 and advance=1, longer windows report better or equal objective function values than shorter windows.

Table 2. For PM, Percent above Monolith's Objective Function Value Achieved by the Cascade Heuristic for Combinations of Window and Advance

	Window=2	Window=3	Window=4	Window=5	Window=6	Window=7	Window=8	Window=9	Window=10	Window=11
Advance=1	4.45%	0.031%	0.514%	0.207%	0.107%	0.266%	0.146%	0.209%	0.000%	0.000%
Advance=2		1.016%	0.199%	0.474%	0.466%	0.088%	0.209%	0.435%	0.000%	0.000%
Advance=3			2.714%	0.312%	0.306%	0.122%	0.336%	0.000%	0.000%	0.000%
Advance=4				0.709%	0.568%	0.477%	0.000%	0.000%	0.000%	0.000%
Advance=5					0.845%	0.000%	0.000%	0.000%	0.000%	0.000%
Advance=6						0.000%	0.000%	0.000%	0.000%	0.000%
Advance=7							0.000%	0.000%	0.000%	0.000%
Advance=8								0.000%	0.000%	0.000%
Advance=9									0.000%	0.000%
Advance=10									0.000%	0.000%

For example, with advance = 1 and window = 2, the heuristic's objective function over-estimates optimal by 4.45%.



The distribution of the PM objective function values, as percent increase from the monolith's objective function value, for each window, when the advance is varied. For all combinations of window and advance not shown, the cascade heuristic objective function value is equivalent to the monolith's objective function value.

Figure 3. For PM, Percent above Monolith's Objective Function Value Achieved by the Cascade Heuristic

For PM, there are no conclusive statements about how long an advance should be for a particular window length. Shorter advances mean that more iterations of the model are being solved in rapid succession, so myopic decisions made in early time periods can be quickly rectified as new information about future periods is encountered. However, the PM cascade results show that shorter advances do not nessessarily equate to better-quality solutions, as we see for a window of seven, where an advance of two yields a higher quality solution than an advance of one. For this model, setting advance = 1 represents the most reliable choice for producing high-quality results. While, for a given window length, there could be a value of advance that yields an even better solution, it is impossible to discern that without testing all possible values of advance.

Another reason for the high-quality solutions from the cascade of PM is that demand is homogeneous over time, as shown in Table 3. The demands listed in Table 3 are used for each cascade of PM. As there are no significant changes to requirements as time progresses, production decisions made in early periods remain good decisions over time. Adding a demand spike in just one period during the later half of the model has a significant impact on solution quality. Specifically, we revise the formula to calculate demand for period 10 to be:

$$demand_{p,t} = \max\{0, D\},$$

where

$$D \sim 3 * N\left(demand\_scale * seasonal_{10}, 0.5 * demand\_scale * seasonal_{10}\right).$$

This modifies the demand for period 10 from 1,103.366 to 10,598.25, 529.688 to 7,495.7439, 2,007.06 to 1,726.069, and 1,032.468 to 11,614.443 for each product, respectively. Solving the model with modified demand using a cascade heuristic with various window lengths and the advance held constant at one, the quality of the objective function reported significantly declines. Figure 4 shows the new objective function values reported for each of the six runs conducted. The objective function value from the cascade heurstic does not fall within 2% of the optimal solution until the window length is seven.

Table 3. Original Demand for PM by Time Period and Product

		Time Period				
Product	**t01**	**t02**	**t03**	**t04**	**t05**	**t06**
p01	350.4967	312.263	1159.84	2390.266	423.3956	1355.011
p02	1331.367	1785.4928	2106.669	1031.413	890.9129	518.9402
p03	1502.197	623.8679	1377.981	2400.298	1421.769	284.731
p04	375.8913	203.0075	593.7956	898.7243	728.2773	1641.388
	t07	**t08**	**t09**	**t10**	**t11**	**t12**
p01	1508.655	637.9648	500.3038	1103.366	1005.408	1366.323
p02	946.2021	868.6966	736.6907	529.6878	3982.034	3532.75
p03	2049.636	1336.6945	1084.476	2007.057	3267.691	4756.55
p04	809.5161	643.2418	1035.47	1032.468	239.4918	2498.581

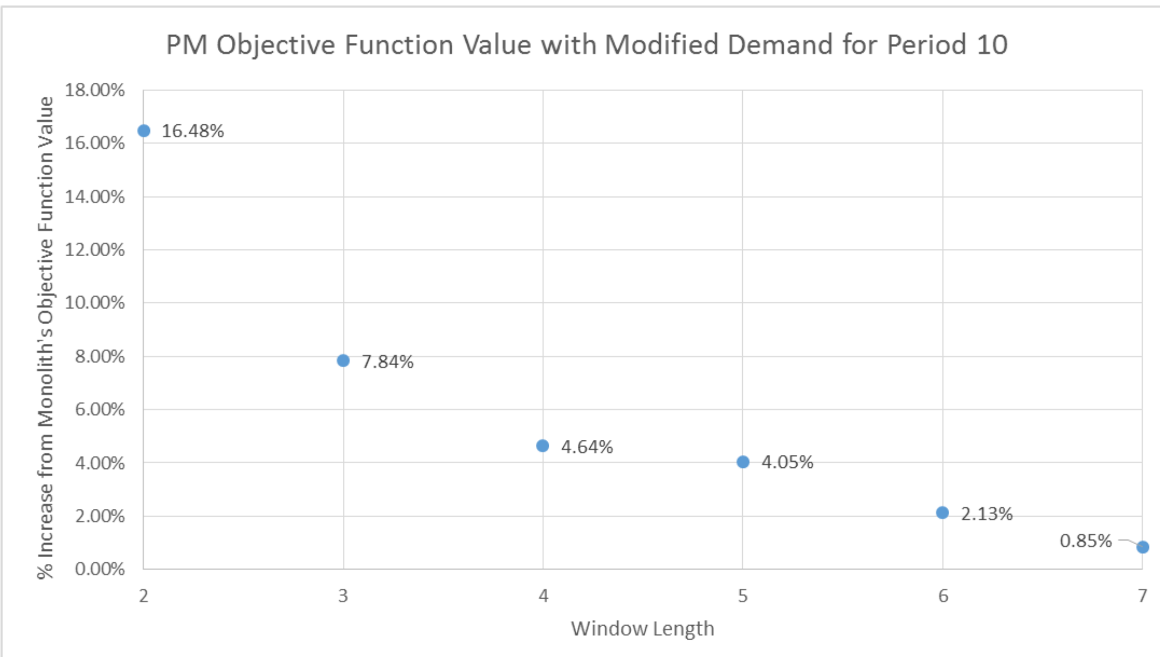


Figure 4. Objective Function Value of PM, as Percent Increase from Objective Function Value, with Modified Demand for Period 10 When Advance=1

B. HASMa CASCADE

For a detailed description of the data included in HASM, see Zerr (2016). This same data is included in HASMa. As HASMa is composed of 15 time periods, 154,696 decision variables and 117,423 constraints, the run time for this model is considerably longer than that of PM. While the HASMa monolith solves in 312 seconds, certain combinations of window and advance take upwards of 5,800 seconds to solve to optimality. As a result, we limit the combinations of window and advance. To first develop a baseline understanding of how the window length effects the objective function value reported, the length of the advance remains fixed at one while the length of the window takes on values 2, 3 … 15. Not until the window includes more than two-thirds of the periods in the model does the objective function value fall within 20% of the optimal solution. Figure 5 shows how a window length of eight marks the first significant improvement in the quality of the objective function value. Henceforth, we only consider window lengths of at least eight.

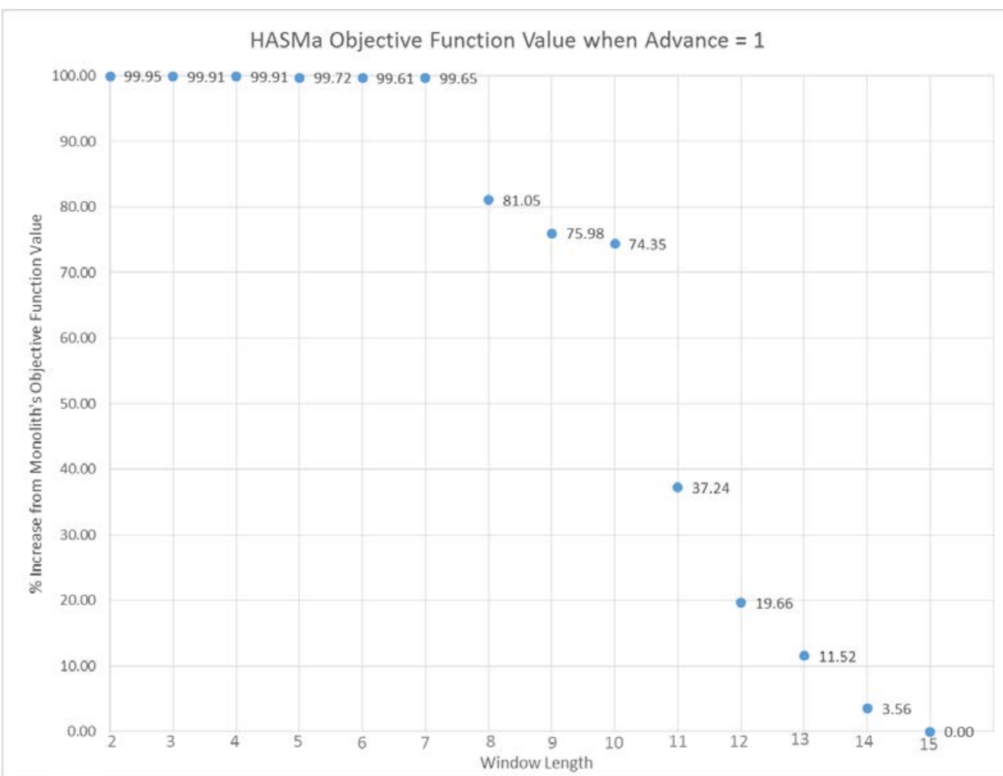


Figure 5. HASMa Objective Function Values for Cascade When Advance = 1

The quality of solutions produced by the cascade applied to HASMa is quite different from the quality of solutions produced in the cascade heuristic with PM. This results from the inability of HASMa to easily recover from poor decisions made in early time periods. The maintenance depots can accommodate only a fixed number of aircraft at any particular time. Shorter window lengths place priority on achieving flight hour requirements in the early periods. As a result, not enough aircraft are scheduled for service life extensions, and in the later time periods there are neither enough capable aircraft remaining to meet flight hour goals, nor availability of depot floor space and man-hours to rectify the problem.

Figure 6 shows the degradation of solution quality when the window length is held constant and the advance is increased. These findings represent a significant deviation from the findings of PM, where there was no monotonic trend seen between the length of the advance and the quality of the solution. Shorter advances mean that the model requires more iterations to solve in its entirety. While shorter windows can produce myopic solutions, shorter advances can provide the opportunity for the model to rectify poor decisions. For example, setting the window = 10 and advance = 5 means that despite regarding two-thirds of the periods in the first window, the later time periods of the model are not considered at all until the solutions for the first third of the model have been locked down. Conversely, setting window = 10 and advance = 1 means that after executing just one year’s worth of aircraft assignments, the model is re-evaluated and later time periods are considered. The price for this improved vision is the increased computation time required to consider long windows with short advances.

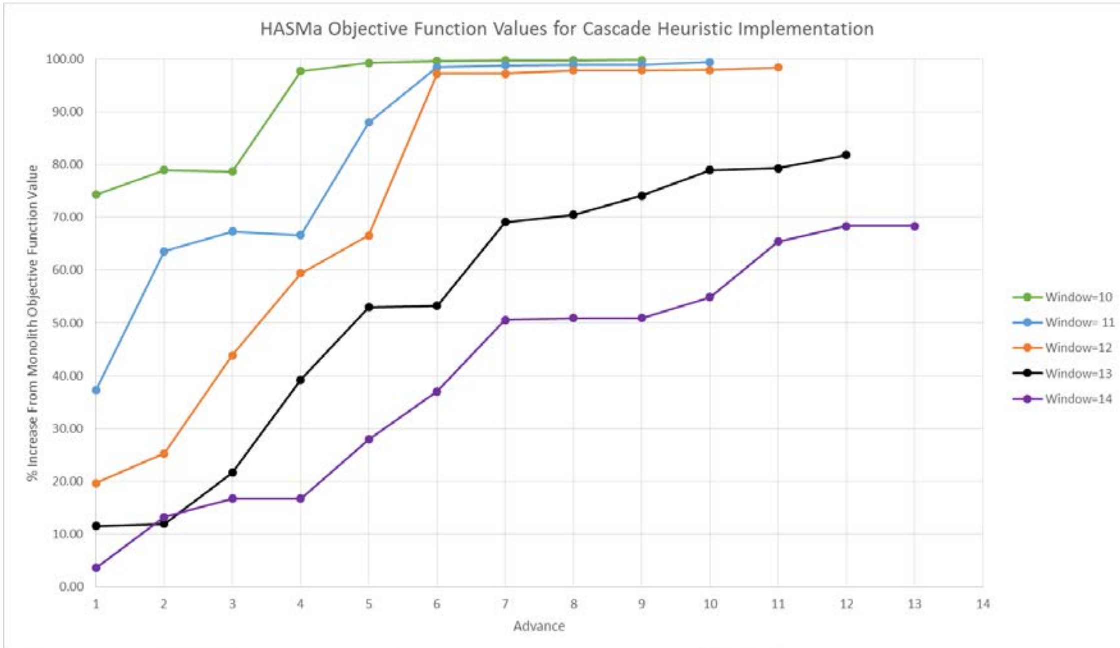


Figure 6. HASMa Objective Function Value for Cascade Heuristic When Window= 10, 11,…,14 and Varied Advances

PM is a cost-minimization model where the objective function value reported represents the lowest possible cost incurred while meeting the supply and demand requirements of each period. The objective function value of HASMa, however, incorporates several other components in addition to cost, as the purpose of this model is to meet flight hour requirements for each time period while still completing required maintenance and service life extensions. The elastic penalties included for violating particular constraints heavily influence the value of the objective function for HASMa. Figure 7 shows how, regardless of the window length, each solution fails to meet the flight hour requirements for years one through five, and then again for years 10–14. Whereas, in some years, shorter window lengths struggle to meet flight hour requirements at the end of the model by over 9,000 flight hours, the monolith deviates by no more than 2,400 flight hours from the number required.

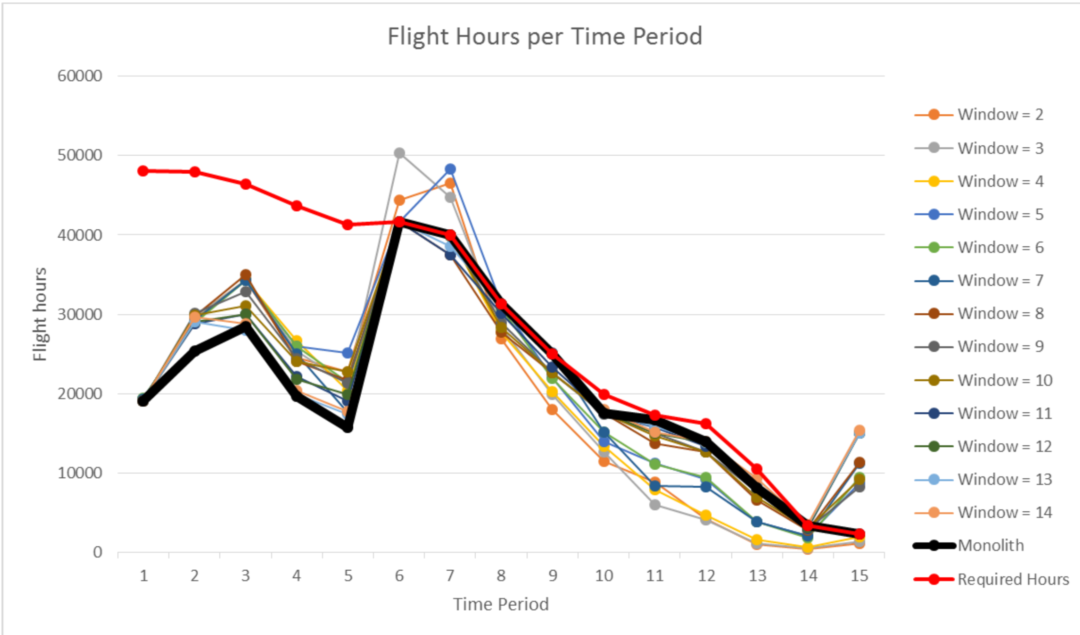


Figure 7. Flight Hours Prescribed per Time Period by Cascade of HASMa When Advance = 1

These results provide evidence of the myopic solutions often produced by short windows. Solved with a short window, particularly one that consists of less than a third of the total periods, HASMa over-prescribes flight hours in the early stages of the model in an attempt to minimize the penalty for violating the required flight hour constraint. The cascade then makes reactionary decisions in the later periods because not enough aircraft have received the SLE and HFH maintenance to allow them to continue to accrue flight hours. This results in high deviations from the flight hour requirement and a higher penalized objective function value. Figure 8 shows how the total number of flight hours prescribed by each solution varies only slightly between window length, with all trials (monolith included) failing to meet the total flight hour requirement. However, the assignment of flight hours within each time period has a drastic effect on the objective function value, and more importantly on readiness of the USMC F/A-18 community. As an attempt to rectify this, additional constraints or elastic penalties can be added to the model that enhance the value of decisions in later time periods, a technique used by Zerr (2016) on HASM.

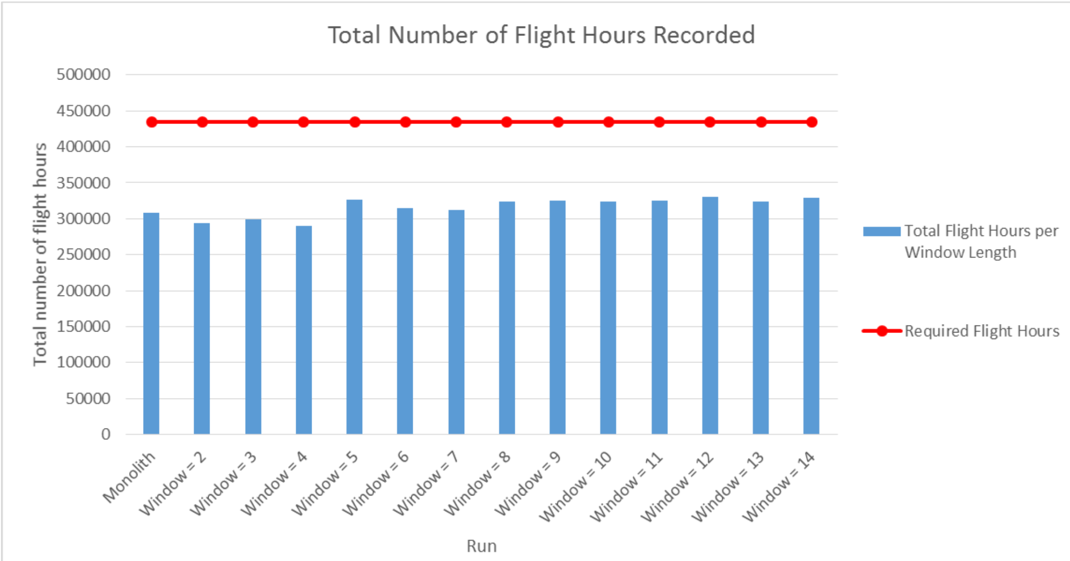


Figure 8. Total Flight Hours Prescribed by Cascade of HASMa by Window Length When Advance = 1

C. PRODUCTION MODEL AGGREGATION

This thesis considers two aggregated versions of PM. Production-1A fixes τ' to one, which results in a single aggregation at the end of the model. The value of τ takes on all possible values between 1, 2…, 11. Figure 9 shows the periods included in the aggregation for each of the 11 runs of the model. Sixty-eight trial runs of Production-2A were conducted, one for each unique combination of τ and τ' where $\tau = 2,3,...,12$ and $\tau' = 1,2,...,11$. Varying the values of τ and τ' allows for comparison of the lower bounds produced when the same size segment of non-aggregated time periods starts at a different time period within the model. For example, Figure 10 shows how, by varying the values of τ and τ', a non-aggregated segment consisting of six periods emerges in six different locations across the model. A comparison of the objective function value that results from each of the six runs indicates that the location of the two aggregations can cause significant variability in the quality of the resultant lower bound. Considering all values of $\tau = 2,3,...,12$ and $\tau' = 1,2,...,11$ allows for the analysis of non-aggregated segments comprised of one to eleven periods.

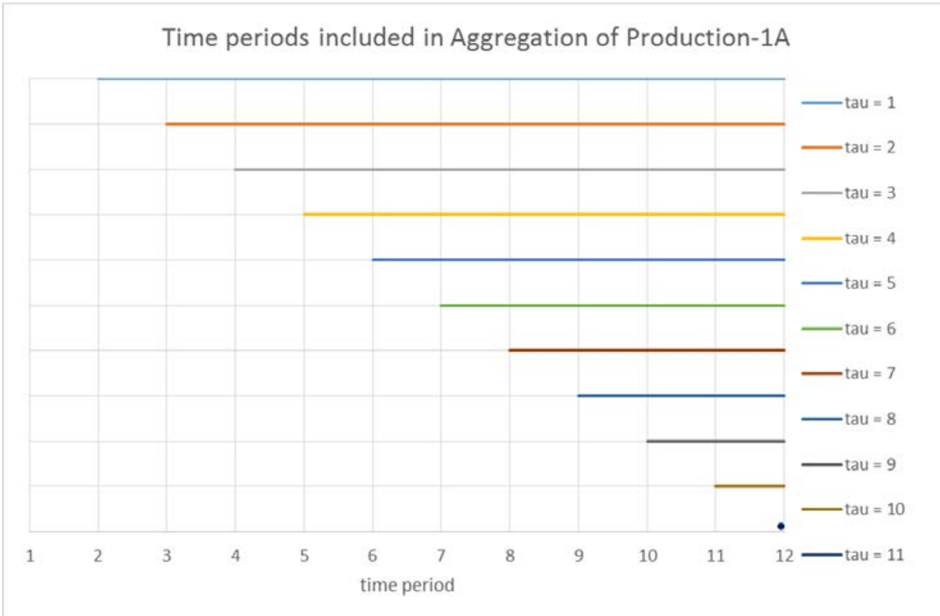
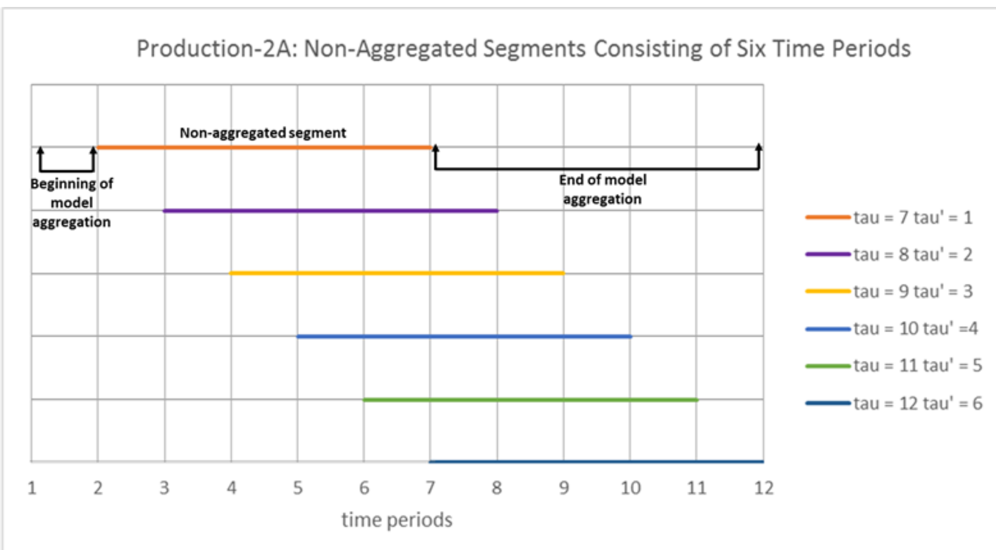


Figure 9. Time Periods Included in Aggregation of Production-1A

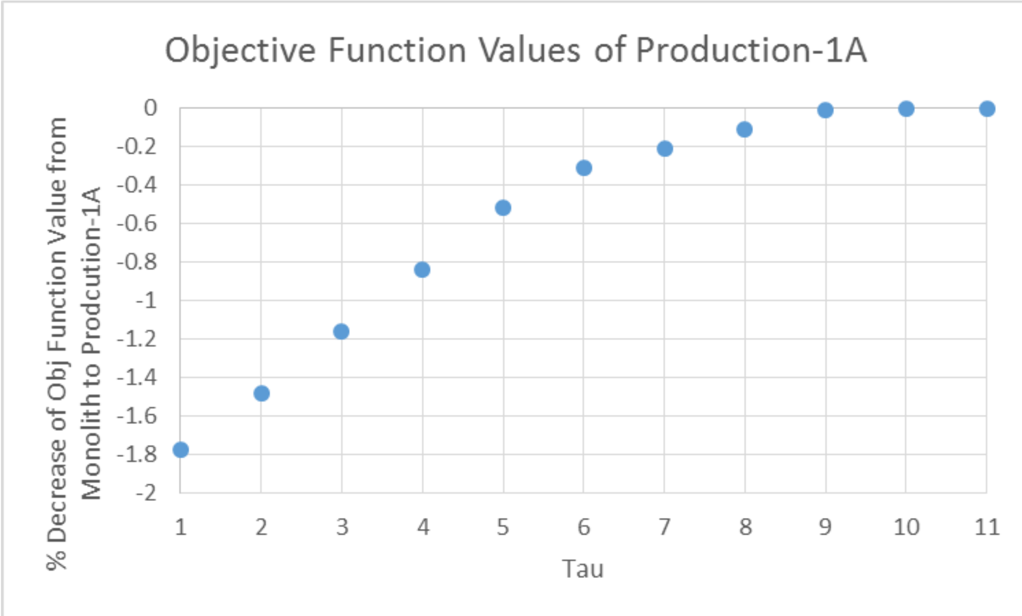


The 68 runs completed of Production-2A produced six non-aggregated segments of size six. The colored segments of the graph show the location of the non-aggregated segments within the larger model. All time periods before the colored segment are part of the early aggregation, while all the time periods at the end of the model are part of the later aggregation. Similar graphs could be produced to show non-aggregated segments of length one through eleven.

Figure 10. The Six Non-aggregated Segments of Length Six for Production-2A

1. Production-1A Analysis

Figure 11 shows the percent decrease of the objective function value reported during the trials of Production-1A from PM's monolith. As the value of τ increases, the percent deviation between the reported objective function value for Production-1A and the monolith decreases. As τ increases, the aggregated model solves more periods under the same conditions as the monolith, and thus this decrease in deviation (increase in solution quality) is expected. For $\tau =$ 9, 10, and 11 the objective function value of the aggregated model is equivalent to the objective function value of the monolith. With the largest possible aggregation, when τ = 1 and all but the first time period is included in the aggregation, there is less than a 2% deviation between the reported objective function value and the monolith. For this particular model, all values of τ produce a lower bound that provides the user of the model the ability to make informed decisions that would result in near-optimal conditions. Without developing an aggregated model, a simple way to produce a lower bound is to solve the model as a relaxed mixed integer program (RMIP), which relaxes the integer requirement on discrete variables. However, it is not always possible to solve the relaxation if the problem is too big. The RMIP solution for PM produces a lower bound of comparable quality to those produced by Production-1A, with its percent deviation from the optimal objective function value at just 0.69%.

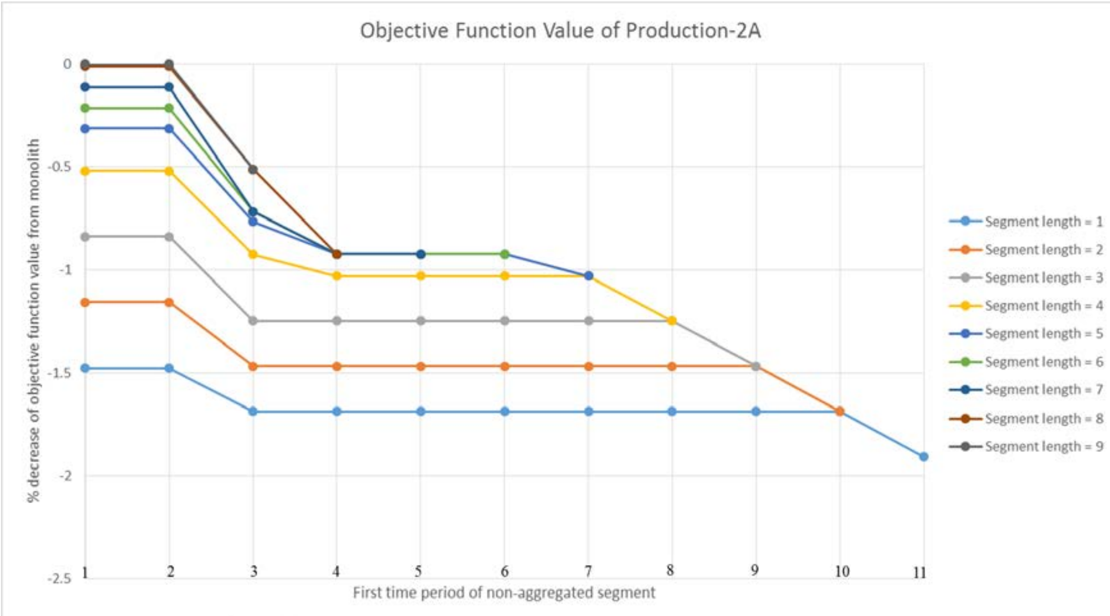


The objective function value of each of the 11 runs completed of Production-1A, represented as the percent decrease from the monolith's objective function value.

Figure 11. Objective Function Values of Production-1A

2. Production-2A Analysis

Figure 12 summarizes the results for the 68 runs of Production-2A. As the number of time periods not included in either aggregation increases, the deviation of the objective function value from the monolith decreases, which improves the lower bound produced by the aggregation. However, even when only one period is included in the non-aggregated segment, the lower bound is still within 2% of the optimal objective function value. As the lower bounds produced by Production-1A were all within 2% of the optimal objective function value, the results from Production-2A do little to improve them. Figure 12 also shows the relationship between the location of the segment of non-aggregated periods and the quality of the bound produced. Consistently across all non-aggregated segment lengths, the quality of the lower bound improves as the segment moves earlier in the model.



Production-2A objective function values as the percent decrease from the monolith's objective function value. Series on the plot show the various lengths of the non-aggregated segment of periods, while the horizontal axis shows the location of the non-aggregated segment.

Figure 12. Production-2A Objective Function Values, Shown as the Percent Decrease from the Monolith's Objective Function Value

D. HASMa AGGREGATION

This thesis also considers two aggregated versions of HASMa. HASM-1A fixes τ' to one so only one aggregation occurs at the end of the model, while the parameter τ takes on values of $1, 2, ..., 14$ for each of the 14 different runs of HASM-1A. HASM-2A considers all possible combinations of τ and τ' where $\tau = 2, 3, ..., 14$ and $\tau' = 1, 2, ..., 13$, which results in 91 trial runs.

1. HASM-1A Analysis

Figure 13 shows the percent decrease of the objective function value of HASM-1A from the objective function value of the HASMa monolith. Similar to the results found with Production-1A, as the value of τ increases, the percent deviation between the objective function values from each trial of HASM-1A to the monolith's objective function value decreases. For $\tau =$ 13 and 14 is there no deviation between the objective function value of HASM-1A and the HASMa monolith. With the smallest aggregation,

when τ = 1, there is a more than 50% deviation between objective function values. Not until the aggregation includes more than two thirds of the total periods of the model does the lower bound fall within 10% of the optimal objective function value. While the general trend of the results is in keeping with the results found with PM, the quality of the lower bounds produced by HASM-1A is significantly worse than those produced by Production-1A, where even the worst lower bound reported is within 2% of the optimal objective function value. These findings also mirror the results of the HASMa cascade solutions, where setting the window length at 10 marks a significant improvement in the quality of the objective function value. For HASM-1A, setting τ = 10 marks the first significant improvement of the lower bound.

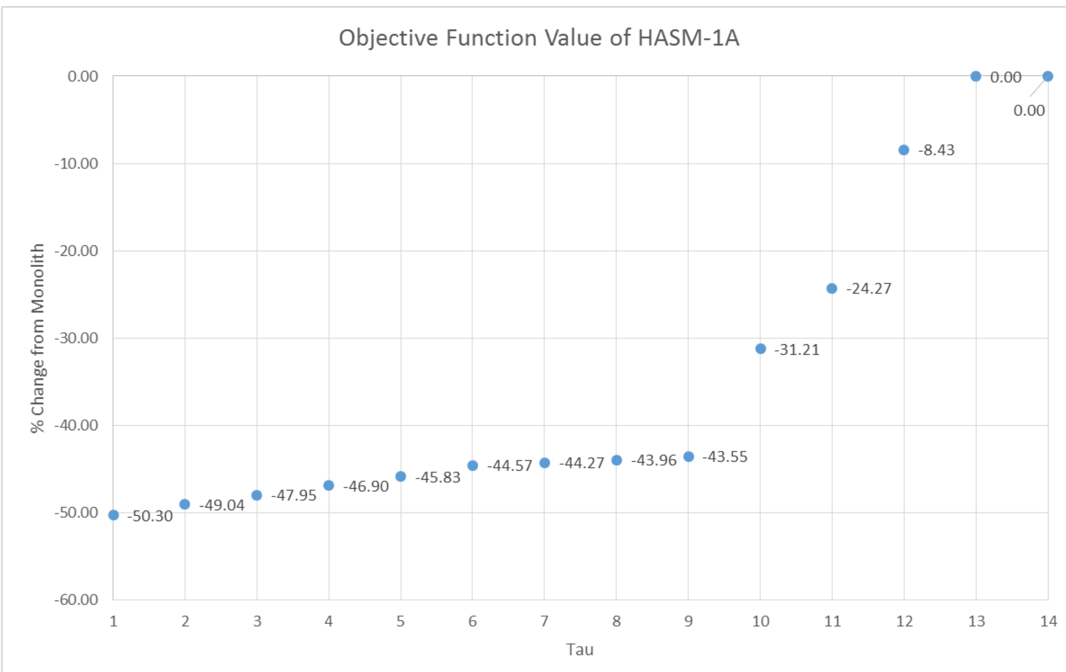


Figure 13. HASM-1A Objective Function Values as the Percent Decrease from the Monolith's Objective Function Value

In conjunction with the lower bound produced by HASM-1A, we produce a corresponding upper bound for the model by solving HASM-1A using a cascade. For example, setting $\tau = 6$ produces a lower bound for the objective function value of 362,460, a 44.57% decrease from the optimal objective function value. We then solve

HASM-1A, keeping $\tau = 6$ and setting the window=5 and the advance=2. This produces an upper bound for the model that is a 99.79% increase from the optimal objective function value. Table 4 summarizes the results from this analysis. The upper bounds produced with this method are only slightly better than the cascade results for the same window and advance without the aggregation. However, by developing upper and lower bounds for the model, we are able to see, even without any comparison to the monolith's solution, the potential for low-quality solutions produced by a cascade. Exploring the full range of upper and lower bounds, for all values of τ, could provide insight into the range of window and advance to consider when implementing a cascade heuristic on the original model.

Table 4. Upper Bounds Produced for HASMa by Solving HASM-1A with a Cascade Heuristic

window	advance	tau	% Deviation
5	2	6	99.791
6	3	7	99.588
7	4	8	94.573
8	5	9	97.927
9	6	10	97.424
10	2	11	64.677
11	2	12	62.041

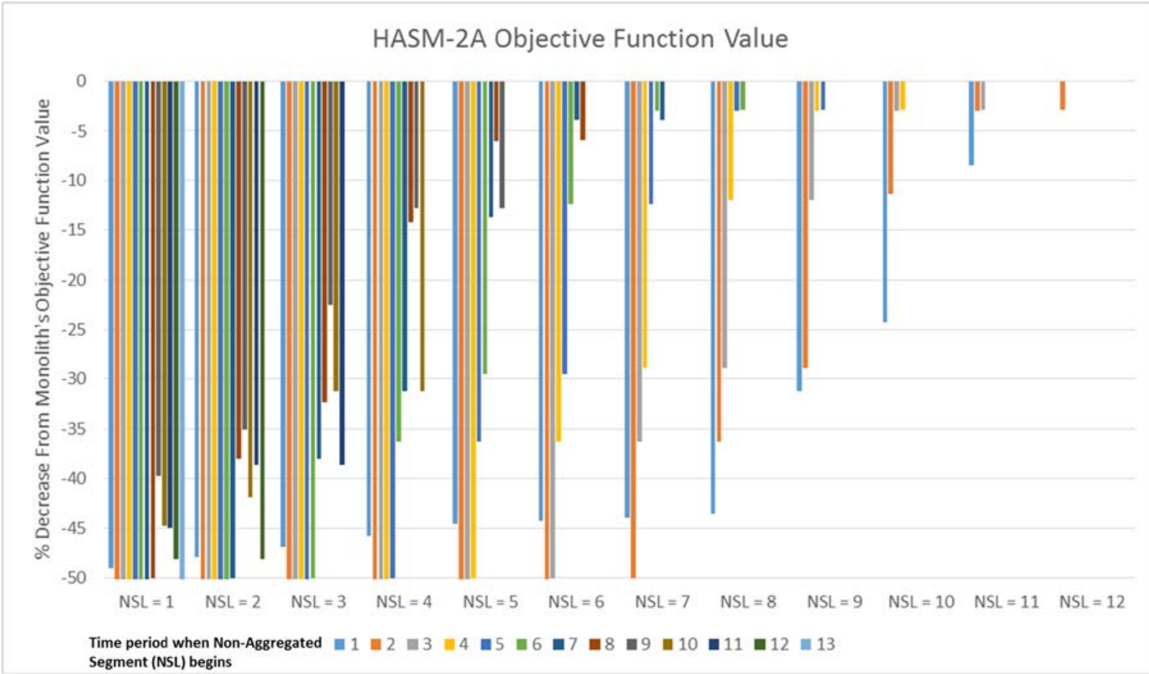
2. HASM-2A Analysis

When directly comparing the results of HASM-2A to HASM-1A for any given value of τ, adding an additional aggregation at the beginning of the model produces a worse quality lower bound. For example, when $\tau = 8$ HASM-1A produces a lower bound that is a 43.55% decrease from the monolith's objective function value. When $\tau = 8$ and $\tau' = 2$ the objective function value for HASM-2A is a 50.48% decrease from the monolith. This holds true when comparing all objective function values from HASM-1A and HASM-2A in a similar manner. However, HASM-2A does improve the quality of the lower bound when comparing the objective function values produced by the same number of non-aggregated periods in each model. When $\tau = 8$ there are eight non-

aggregated periods in HASM-1A, and the lower bound is a 43.55% deviation from the monolith. For HASM-2A, when there are eight non-aggregated time periods, the percent deviation from the monolith varies from a 43.55% deviation to a 2.91% deviation for various values of τ and τ', as shown in Table 5. Figure 14 displays the results for each run of HASM-2A, while Figure 15 shows only the results from HASM-2A that are within 15% of the optimal objective function value. HASM-1A does not produce a lower bound within 10% of the optimal objective function value until $\tau = 12$, where the model includes just three periods in the aggregation. Adding a second aggregation to the beginning of the model results in lower bounds within 10% of the optimal objective function value with the exclusion of as few as ten periods from the aggregation. The quality of all lower bounds produced by HASM-1A and HASM-2A is significantly better than that produced by the HASMa RMIP solution, which deviates from the optimal objective function value by 60.88%.

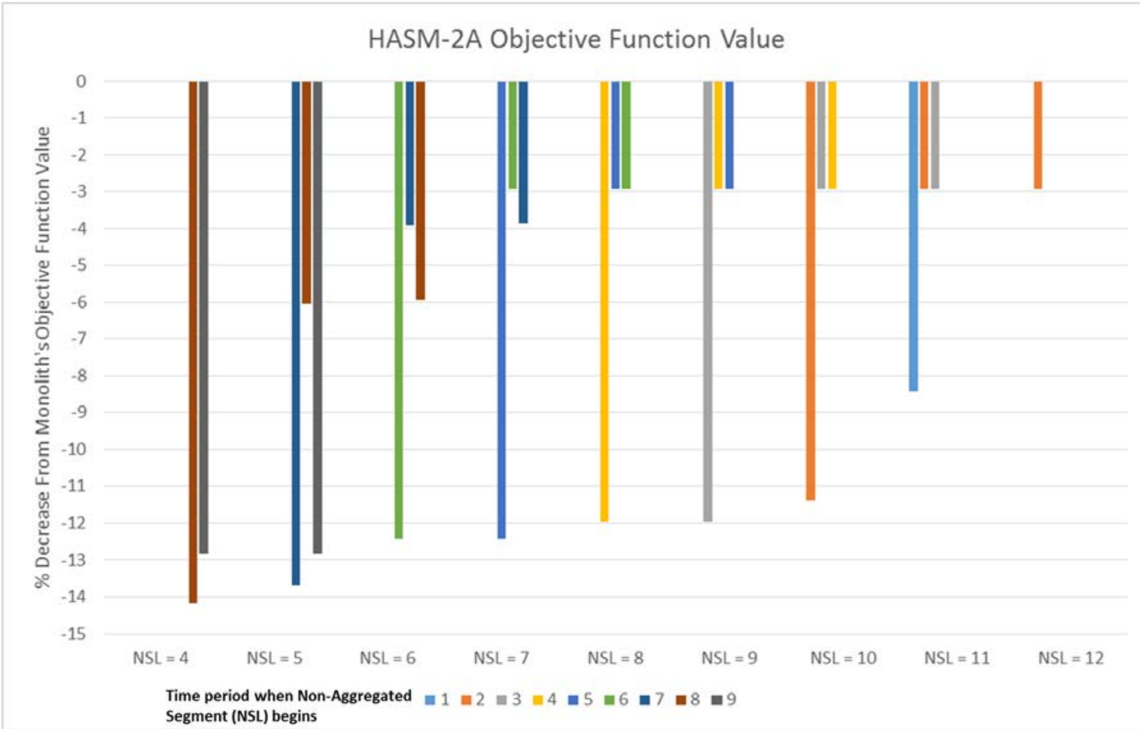
Table 5. HASM-2A Objective Function Values, as the Percent Deviation from the Monolith for a Non-aggregated Segment Length of Eight

Non-Aggregated Segment Length = 8		
tau	**tau'**	**% Deviation**
9	1	-43.547
10	2	-36.240
11	3	-28.898
12	4	-11.967
13	5	-2.938
14	6	-2.913



HASM-2A objective function values as the percent decrease from the monolith's objective function value. The clusters of bars represent the multiple lower bounds for each non-aggregated segment length (NSL). The color of the bar represents the first time period of the non-aggregated segment.

Figure 14. HASM-2A Objective Function Values as the Percent Decrease from Monolith's Objective Function Value



HASM-2A objective function values as the percent decrease from the monolith's objective function value. The clusters of bars represent the multiple lower bounds for each non-aggregated segment length (NSL). The color of the bar represents the first time period of the non-aggregated segment. This graph shows the subset of lower bounds that are within 15% of the optimal objective function value.

Figure 15. HASM-2A Objective Function Values as the Percent Decrease from Monolith's Objective Function Value for a Subset of Runs

While the results of Production-2A display a monotonically decreasing quality of lower bound when the non-aggregated segment moves to later periods, this trend only occurs for the HASM-2A solutions when the non-aggregated segment includes more than seven time periods. For non-aggregated segments of seven and less, there is an initial decrease in lower bound quality as the non-aggregated segment moves later in the model. However, the lower bound quality then begins to improve as the non-aggregated segment enters the middle periods of the model, before again decreasing when at the end of the model. For example, when the non-aggregated segment length is four, as shown in Figure 16, the quality of the lower bound begins to improve when the non-aggregated segment begins at the sixth time period and the quality of the lower bound begins to decrease again when the non-aggregated segment begins at the 10th period. While non-aggregated segments of all lengths less than eight display similar characteristics for their

general behavior, the quality of the lower bounds increases and decreases at different time periods for different segment lengths.

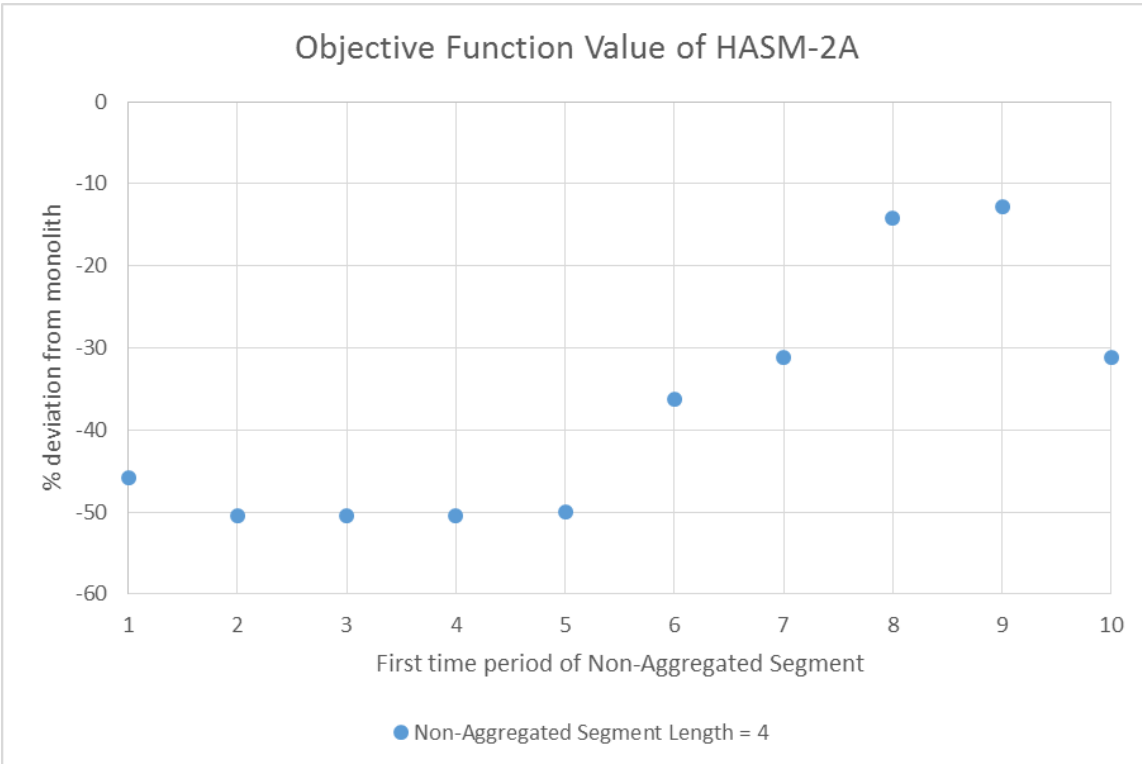


Figure 16. HASM-2A Objective Function Values for Non-aggregated Segment Lengths= 4

We can partially explain these results by recalling the inability to meet flight hour requirements for close to two-thirds of the periods included in the model. The optimal solution produced by the monolith fails to meet the flight hour requirement goal until the sixth year, however during years six through nine the monolith is able to meet flight hour requirements to within two hours. Between years 10–13, the monolith again fails to meet flight hour requirements by a significant amount. Including at least one of these middle periods, when the model has the capability of meeting flight hour requirements, in the non-aggregated segment appears to be critical to producing a higher quality lower bound.

THIS PAGE INTENTIONALLY LEFT BLANK

V. CONCLUSIONS AND RECOMMENDATIONS

A. CONCLUSIONS

This thesis considers two separate applications to study the effects on solution quality of changing the length of the window and advance when implementing a cascade heuristic. We chose to study PM because of its simplicity and general applicability to other ILPs. We chose to study HASMa for both its real-world application and the additional complexity added when objective functions include elastic penalties. This thesis then goes on to implement aggregations of each model that produce a lower bound for the optimal objective function value.

The cascade results found from each of the two models are similar in some aspects and vastly different in others, which highlights the complexity of the cascade technique and the difficulty in making any broad conclusions regarding the application of a cascade heuristic. The cascade results for each model support choosing the longest window length that is both time and computationally feasible to produce higher-quality solutions for any cascade. This thesis produces less conclusive evidence to back a statement regarding the length of the advance. Results from HASMa suggest that shorter advances typically yield higher-quality solutions, but this is not always the case for PM. We can make no specific statements regarding how long the window length should be with respect to the total length of the model to guarantee any quality of solution.

The quality of the solution produced using a cascade is highly dependent upon the data, as is seen in both models. The cascade solutions for HASMa are poor until enough time periods are included in the first window to see nearly to the end of the model. We see the dependence of solution quality upon data again in PM when we modified the demand in period 10 to reflect a significant increase from previous periods. Models with the ability to recover quickly from decisions made in early periods produce solutions that are more resilient to variations of the length of the window and advance.

Baker (1997) suggests that the advance between windows should be, “at least as large as the maximum number of time period indices that are common to consecutively

indexed rows" (Baker, 1997, p. 6). Setting the length of advance to one for HASMa violates this recommendation, as there is an overlap between consecutive time periods. All maintenance events, save for PMI, take more than one period to complete, so an aircraft assigned to HFH in period two must remain there beyond the start of period three. However, for each window length, we found that setting the advance to one produced the highest-quality solutions.

For both models examined the aggregation produces a bound for the model that, when paired with the results from the cascade, provides an unambiguous measure on the quality of the cascade. Obtaining the HASMa results that show the huge deviation between the cascade solution and the lower bound clearly reveals the necessity to change the implementation of the heuristic to produce a more acceptable deviation. On the other hand, comparing the solutions obtained by each method for PM indicates that any length of window and advance should provide a solution within 5% of the true optimal solution. The results from this thesis suggest the importance of determining bounds for any model solved using a cascade. For PM, the lower bounds produced through aggregation are comparable to the lower bound produced by the RMIP relaxation of the model. However, the lower bounds produced by HASM-1A and HASM-2A are all significantly better than the lower bound produced by the RMIP approximation to HASMa.

Comparing the results of each technique used to solve HASMa provides valuable information about the significance of the elastic penalties associated with different periods of the model. In order to combat the end effects produced by the six-month window length implemented, Zerr (2016) weights the objective function of HASM to ensure the model meets flight hour requirements in later periods. However, this results in a drastically inflated objective function value for cascades with short window lengths that fail to meet flight hour requirements in later periods. In reality, though longer windows do a better job meeting flight hour requirements across all periods, the deviation between results produced by short and long window lengths is not nearly as dramatic as the objective function value makes it seem. For the cascade results, setting the window length to more than ten results in significantly improved findings, in part, because the aircraft assignments made in the early periods include the critical information about

periods six through ten. The lower bounds produced in the aggregations of the model improve significantly when the set of critical periods is included in the non-aggregated segment and solved under the same constraints as the monolith.

B. RECOMMENDATIONS FOR FUTURE WORK

There are many different aspects of implementing a cascade heuristic that could be studied in depth. This thesis used only two models to study the effect of the length of window and advance on the quality of cascade heuristic solutions. Both models studied in this thesis were finite-horizon ILP applications. By studying a wider variety of models, such as infinite-horizon models, more conclusive statements regarding the optimal length of window and advance might be made. There is also no research conducted in this thesis on non-uniform window lengths across the model. HASMa might benefit from this type of analysis.

Studies of several variations of the PM would enhance the findings of this thesis. One could move beyond using a cascade heuristic for the time horizon of the model and implement a cascade of aggregated stock keeping unit groups. A cyclic cascade, where the heuristic does not begin its computation with the first time period but later in the model, could be used to develop deeper insight into the effect of demand spikes. Finally, a cascade heuristic could be used to seed initial incumbent solutions.

Any future work regarding Zerr's (2016) version of HASM should reference the findings of this thesis. At the time HASM was developed, there was no metric to judge the quality of solution produced by the window and advance chosen by the author when implementing a cascade heuristic. Though the work completed for this thesis uses a simplified version of HASM, the results are nevertheless applicable. Adding an aggregation to HASM to determine a lower bound to the optimal objective function value, and using that bound in conjunction with the results of this thesis, would help ensure the recommendation of a high quality solution to the Marine Corps.

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF REFERENCES

Baker, S. F. (1997). *A cascade approach for staircase linear programs with an application to Air Force mobility optimization* (Doctoral dissertation). Retrieved from Calhoun: http://calhoun.nps.edu/bitstream/handle/10945/8051

Brown, G. G., & Dell, R. F. (2016). Cascade optimization (Working paper).

Brown, G. G., Graves, G. W., & Ronen, D. (1987). Scheduling ocean transportation of crude oil. *Management Science*, *33*(3), 335–346.

Brown, G., Keegan, J., Vigus, B., & Wood, K. (2001). The Kellogg company optimizes production, inventory, and distribution. *Interfaces*, *31*(6), 1–15.

Cascade. (n.d.). In *Oxford English Dictionary*. Retrieved May 22, 2017, from https://en.oxforddictionaries.com/definition/cascade

Dantzig, G. B. (1959). On the status of multistage linear programming problems. *Management Science*, *6*(1), 53–72.

GAMS (n.d.). Retrieved April 11, 2017, from http://www.gams.com

Grinold, R. C. (1983). Model building techniques for the correction of end effects in multistage convex programs. *Operations Research*, *31*(3), 407–431.

Manne, A. S. (1992). Unequal time intervals in economic growth models. *Indian Economic Review*, *27*, 113–121.

Mercenier, J., & Michel, P. (1994). Discrete-time finite horizon approximation of infinite horizon optimization problems with steady-state invariance. *Econometrica: Journal of the Econometric Society*, 635–656.

Miller, B., Alexander, R., Arbige, V., Dell, R., Kremer, S., McClune, B., Oppenlander, J., & Tomlin, J. (2017). Optimal allocation of students to naval nuclear power training units. To appear in *Interfaces.*

Nielsen, L. K., Kroon, L., & Maróti, G. (2012). A rolling horizon approach for disruption management of railway rolling stock. *European Journal of Operational Research*, *220*(2), 496–509.

Walker, S. C. (1995). *Evaluating end effects for linear and integer programs using infinite-horizon linear programming.* (Doctoral dissertation). Retrieved from Calhoun: http://calhoun.nps.edu/handle/10945/35092

Workman, P. E. (2009). *Optimizing security force generation*. (Master’s thesis). Retrieved from Calhoun: http://calhoun.nps.edu/handle/10945/4708

Zerr, G. T. (2016). *Optimization of USMC Hornet inventory*. (Master's thesis). Retrieved from Calhoun: http://hdl.handle.net/10945/49419

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California